République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: Modèle Intelligent et Décision(M.I.D)

Implémentation des algorithmes de traitement des images et vidéos en utilisant la bibliothèque OpenCV

Réalisé par :

- **M^elle kada Fouzia**
- **M^elle zahaf Ahlem**

Présenté le 05 octobre 2016 devant le jury composé de .

- **M^r BENAMAR .A** (Président)
- **M^r .BENAISSA .M** (Encadreur)
- **M^r BENZIEN .Y** (Examinateur)
- **M^r MANA .M** (Examinateur)

Année universitaire : 2015-2016

Remerciements

Louange à الله*, seigneur de l'univers.*

Je remercie toutes les personnes ayant contribué et faciliter la réalisation de ce travail dans de bonnes conditions.

Je remercie plus particulièrement.

Ma famille et mes amis pour l'amour et le soutien qu'ils m'ont

toujours accordés

Monsieur Benaissa Mohamed qui il m'a guidé avec grande Résignation tout au long de la préparation de ce travail. Merci pour tous les conseils, et les bienveillances.

Je remercie vivement les membres de jury, d'avoir accepté de juger mon travail.

Je remercie à l'ensemble des personnels du département de l'informatique à l'université Abou Berk Belkaid.

Merci à tous et à toutes

Dédicaces

-mes parents

-Ma famille

-Tous ceux que j'aime

-Tous ceux qui m'ont aidé

A

FOUZIA

Résumé

Le traitement d'images et vidéo est l'ensemble des méthodes et des techniques utilisées pour améliorer le contenu et la qualité d'une image ou vidéo ou d'en extraire de l'information.

L'objectif de notre travail est d'implémenter les différents algorithmes de traitement d'image en utilisant la bibliothèque OpenCV.

Mots clés : OpenCV, filtrage, histogramme, segmentation, détection de contour, traitement d'image.

Abstract

The images and video processing is the set of methods and techniques used to improve the content and quality of an image or video or to extract information.

The aim of our work is to implement various image processing algorithms using the OpenCV library.

Keys words: OpenCV, filtering, histogram, segmentation, edge detection, image processing.

ملخص

تجهيز الصورو الفيديو هو مجموعة من الأساليب والتقنيات المستخدمة لتحسين المحتوى وجودة صورة أو فيديو لاستخراج المعلومات.

الهدف من عملنا تنفيذهو مختلف خوارزميات معالجة الصور باستخدام مكتبة برمجية مفتوحة للرؤية الحاسوبية(opencv)

كلمات البحث : مكتبة برمجية مفتوحة للرؤية الحاسوبية, الترشيح ,الرسم ,البياني تجزئة ,الكشف عن الحافة ,معالجة الصور

Table des matières

Chapitre III : Traitements des images et des vidéos

Chapitre IV : Implémentation et Résultats

Listes des figures

Chapitre I

Chapitre II

Chapitre III

Chapitre IV

Liste des tableaux

Introduction Générale

A peine plus d'un siècle nous sépare de la naissance du cinéma. Une invention fantastique qui fut rendue possible grâce au développement de la photographie. Depuis cette époque, la technologie de l'image n'a cesse d'´evoluer et de se perfectionner. Alors que la télévision analogique vit ses dernières années en France, les vidéos numériques sont accessibles depuis nos téléviseurs, nos ordinateurs et même depuis nos téléphones portables.

La télévision numérique migre déjà vers la télévision numérique haute définition. La vidéo numérique en général devient un contenu presque aussi commun sur internet que les images numériques. De nos jours, les images et les vidéos numériques sont omniprésentes et la quantité de données associées est gigantesque.

Ces évolutions ont nécessite le développement d'un bon nombre de techniques de traitement de l'image et la vidéo.

Le traitement d'images et vidéo est l'ensemble des méthodes et des techniques utilisées pour améliorer le contenu et la qualité d'une image ou vidéo ou d'en extraire de l'information.

L'objectif de notre travail est d'implémenter les différents algorithmes de traitement d'image en utilisant la bibliothèque OpenCV.

Par voie de conséquence, Notre mémoire est structuré comme suite :

- **Le premier chapitre :** nous parlons de généralités sur la bibliothèque OpenCV. Dans le présent chapitre, nous décrivons cette bibliothèque en donnant son Présentation, sa structure et sa configuration dans le système d'exploitation Microsoft Windows avec Visual Studio 2008.
- **Le deuxième chapitre :** dans ce chapitre nous introduisons brièvement un ensemble des définitions sur les formats d'images et les vidéos.

- **Le troisième chapitre :** ce chapitre est consacré aux différentes techniques de traitement d'image et vidéo.

- **Le quatrième chapitre** : En ce qui concerne le quatrième chapitre nous présentons l'environnement de développement utilisé et notre logiciel dans lequel nous avons présenté Implémentation de quelques algorithmes de traitement d'image et vidéo (segmentation, filtrage, détection de contour) en utilisant la bibliothèque OpenCV.

Chapitre I

Présentation d'Open-CV

1. Introduction

Pour la réalisation de notre projet " le traitement des données visuelles (images et vidéos) " on a besoin d'avoir quelques logiciels ainsi que quelques appareils qui vont être utilisés pour le traitement des données visuelles.

Ce chapitre est consacré à donner une idée générale sur les notions de base de la bibliothèque OPEN-CV et sa configuration sous Windows avec Microsoft Visual Studio 2008.

2. Présentation d'Open-cv

Open-CV (Open Source Computer Vision) est une librairie de traitement et analyse d'images et vidéos avec des interfaces pour les principaux langages de programmation C, C++, Java, C#, Python ...

Cette librairie publiée sous une licence BSD est gratuite pour un usage scolaire ou commercial. Son intérêt est de réaliser des traitements sans forcément connaître les algorithmes. La bibliothèque possède plus de 500 algorithmes optimisés.

Elle a l'avantage d'être optimisée pour les applications temps réelles, de fournir une API bas et haut niveau ainsi qu'une interface pour le langage de programmation parallèle et ces algorithmes stables et puissants[1][5][7][10][11].

2.1. Fonctionnalités

La bibliothèque Open-CV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques.

2.1.1. Traitement d'images

Ce qui concerne le traitement d'image, Open-CV propose des opérations suivant :

- la manipulation d'images (chargement, sauvegarde, copie, conversion...)
- Le traitement d'images (filtrage, détections de discontinuités, morphologie mathématique...)

- l'analyse d'images (composantes connexes, ajustement de primitives...) L'interface graphique (affichage d'images, de vidéos, gestion des évènements...)
- calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs.
- lissage, filtrage.

2.1.2. Traitement Vidéos

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que :

- La lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra)
- La détection de droites, de segment et de cercles par Transformée de Hough
- La détection de visages par la méthode de Viola et Jones
- La détection de mouvement, historique du mouvement
- La détection de points d'intérêts
- La manipulation et acquisition de vidéos
- La vision (calibration de caméra, stéréovision, recherche d'association...)
- Les manipulations de matrices et algèbre linéaire

3. structure de la librairie Open-CV

- ***Highgui :***

Celle-ci permet le traitement des vidéos en temps réel. En voici, ses applications :

- Structures élémentaires
- matrices, tableaux, listes, files, graphes, arbres...
- opérateurs standards sur ces structures,
- Dessin de primitives Géométriques
- lignes, rectangles, ellipses, polygones... et texte.
- Manipulation des images et des séquences
- lecture, écriture...
- Interface utilisateur
- fenêtre, entrées/sorties utilisateur.

- ***CORE :***

Cette bibliothèque permet de :

- manipuler les structures de base
- réaliser des opérations sur des matrices ;
- dessiner sur des images.
- sauvegarder et charger des données dans des fichiers XML…

❖ **Vidéo** : traitement de flux vidéo.

Ces fonctions servent à segmenter et suivre les objets en mouvement dans une vidéo.

❖ **Imgproc**: traitement d'image. Nous entrons dans le cœur du sujet. Les fonctions et structures de ce module ont trait aux transformations d'images, au filtrage, à la détection de contours, de points d'intérêt…

❖ **Features2d**: descripteurs.

Ce module concerne principalement l'extraction de descripteurs selon deux approches courantes (SURF et Star Detector).

❖ **Objdetect**: détection d'objets.

Cette bibliothèque permet de faire de la reconnaissance d'objets dans une image au moyen de l'algorithme Adaboost (Viola & Jones, 2001) .

❖ **Calib3d :** calibration, estimation de pose et stéréovision.

Ce module contient des fonctions permettant de reconstruire une scène en 3D à partir d'images acquises avec plusieurs caméras simultanément.

❖ **ml** et **flann** : classification, regroupement (*clustering*)

4. Installation et configuration avec Visual Studio 2008

4 .1. Visual Studio 2008

Visual Studio est un ensemble complet d'outils de développement permettant de générer des applications web ASP.NET, des services web XML, des applications bureautiques et des applications mobiles.

Visual Basic, Visual C++, Visual C# utilisent tous le même environnement de développement intégré (IDE), qui leur permet de partager des outils et facilite la création de solutions faisant appel à plusieurs langages. Par ailleurs, ces langages permettent de mieux tirer parti des fonctionnalités du framework .NET, qui fournit un accès à des technologies clés

simplifiant le développement d'applications web ASP et de services web XML grâce à Visual Web Developer.

4.2. Mise en place variables d'environnement

Aller sur :

Menu démarrer ---> clic droit sur ordinateur---> propriétés

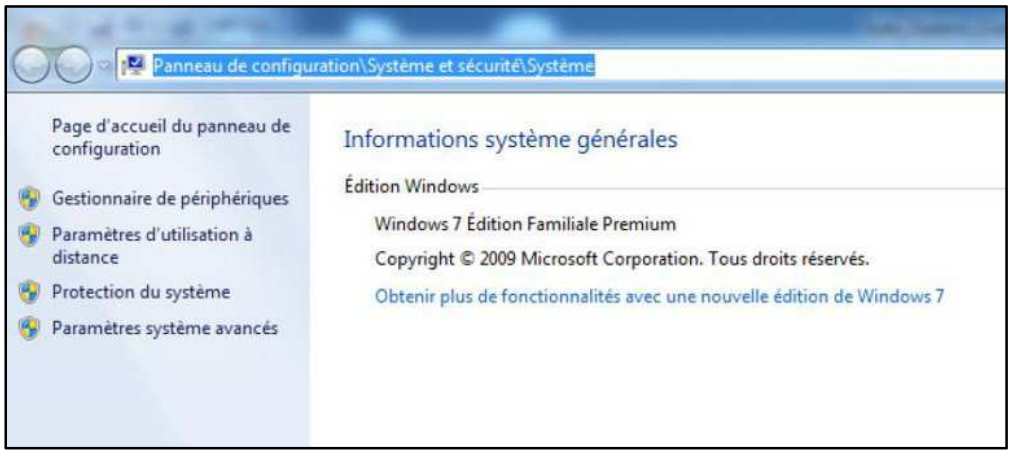


Figure I.1: propriétés du système

- ✓ Choisir Paramètres système avancés (voir Figure I.1)
- ✓ Choisir variables d'environnement … (voir Figure I.2)

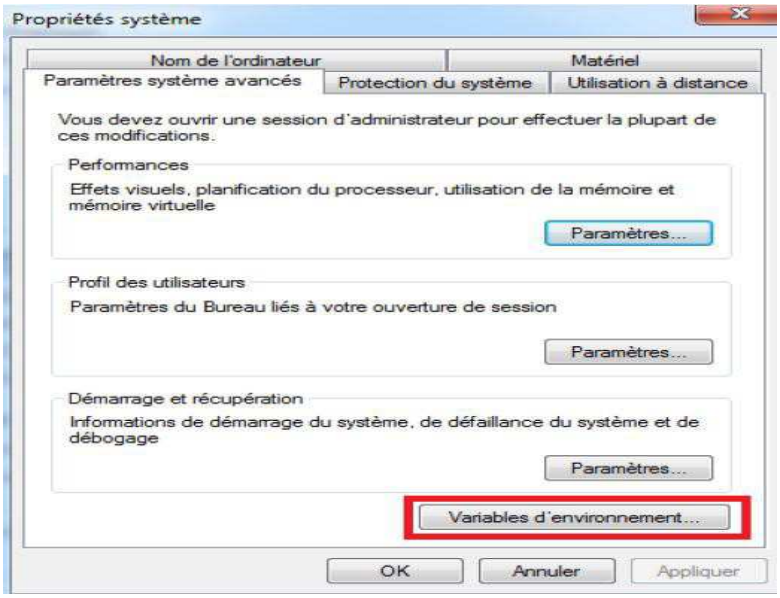


Figure I.2: variables d'environnement

Vous devez ajouter le répertoire « C:\opencv\build\x64\vc09\bin » à votre PATH système. Ce répertoire contient les DLL Open-CV nécessaires pour exécuter votre code.

- Si votre type de système est 32 bits remplacer x64 par x86.
- (C:\opencv\build\x86\VC09 \bin). A noter la partie de VC09 du chemin est la version de Visual Studio installé sur le système (VC09 = Microsoft Visual Studio 2009)

4.3. Configuration de Visual Studio 2008

Il faut procédera deux étapes Pour réaliser la configuration de Visual studio :

- La configuration nécessaire pour chaque nouveau projet.
- La configuration générale, qui est à faire une seule fois.

4.3.1. Configuration générale

Il s'agit d'indiquer à Visual Studio les nouveaux répertoires où sont disponibles les fichiers .h et .lib d'Open-CV.

- Choisir (x64) pour Platform (si le type de système d'exploitation 64 bits) ou bien win32 (si le type de système d'exploitation 32bits).

- Choisir All Configuration pour Configuration.
- Dans le volet de gauche, cliquez sur
- Configuration-----> properties-----> VC++ Directories----->général

- Dans Additional include Directories ajouter les sous répertoires suivants :

  - ✓ C:\opencv\build\include\opencv.
  - ✓ C:\opencv\build\include\opencv2.
  - ✓ C:\opencv\build\include.

Comme le montre la figure I.3

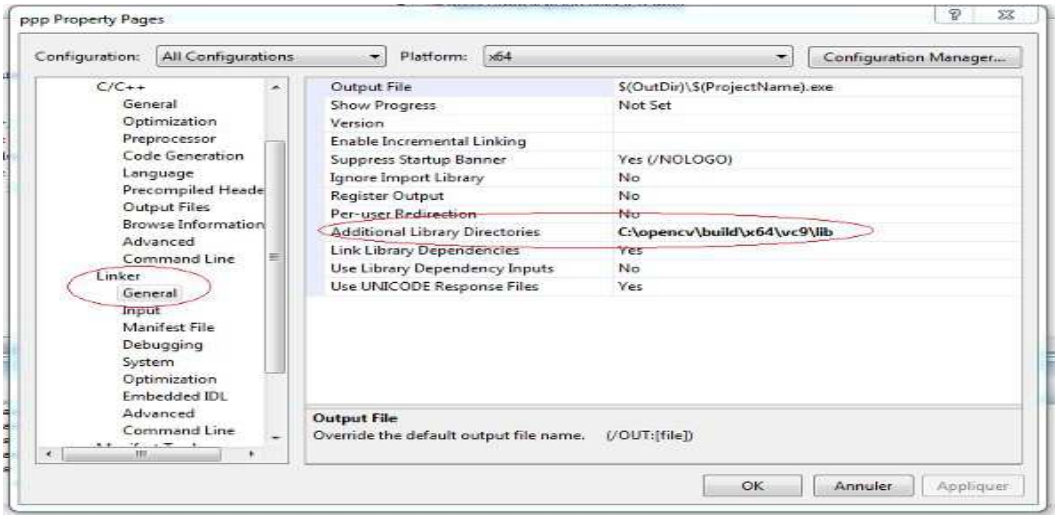


Figure I. 3: Configuration général de Microsoft Visual Studio 2008 pour Open-CV

Remarque : le chemin C:\OpenCV peut varier en fonction de ce que vous avez choisi lors de l'installation.

- Choisir ensuite configuration properties ---> Linker ---> General
- Dans Additional Library Directories ajouter la ligne
  <<C:\OpenCV\build\x64\VC09\lib>> (figure I.4)

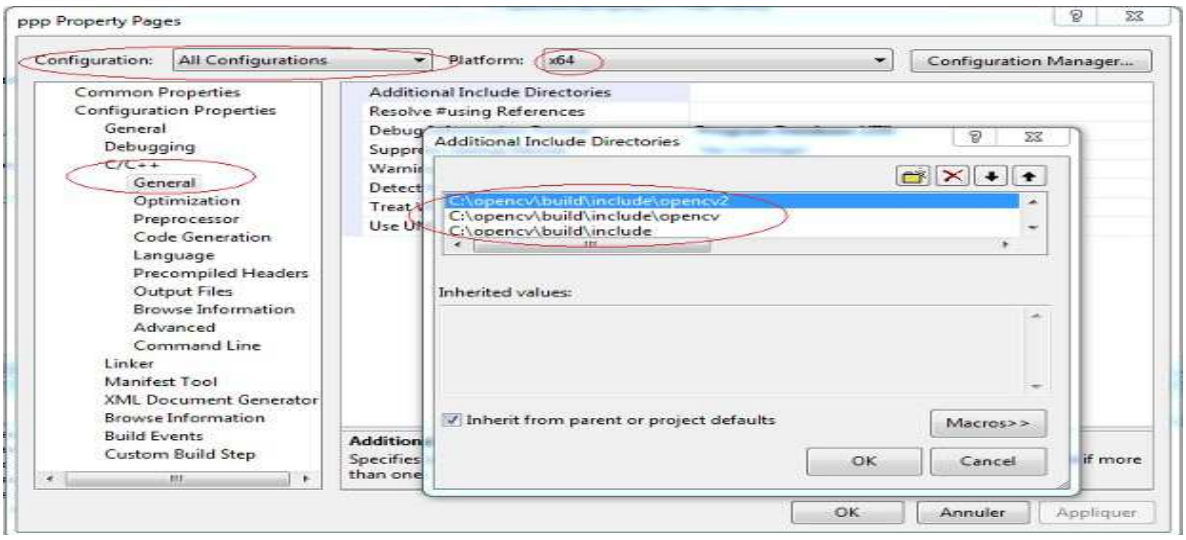


Figure I. 4: Configuration de Microsoft Visual Studio 2008 pour la librairie Open-CV

4.3.2. La configuration nécessaire

Cette étape est à recommencer pour chaque nouveau projet.

- dans configuration properties ---> linker-----> input
- Aller dans Additional Dependencies ajouter les fichiers lib suivants :

  - opencv_core231d.lib
  - opencv_highgui231d.lib
  - opencv_video231d.lib
  - opencv_ml231d.lib
  - opencv_legacy231d.lib
  - opencv_imgproc231d.lib
  - opencv_calib3d231d.lib
  - opencv_contrib231d.lib
  - opencv_features2d231d.lib
  - opencv_flann231d.lib
  - opencv_gpu231d.lib
  - opencv_haartraining_engined.lib
  - opencv_objdetect231d.lib
  - opencv_ts231d.lib

Comme le montre la Figure I.5

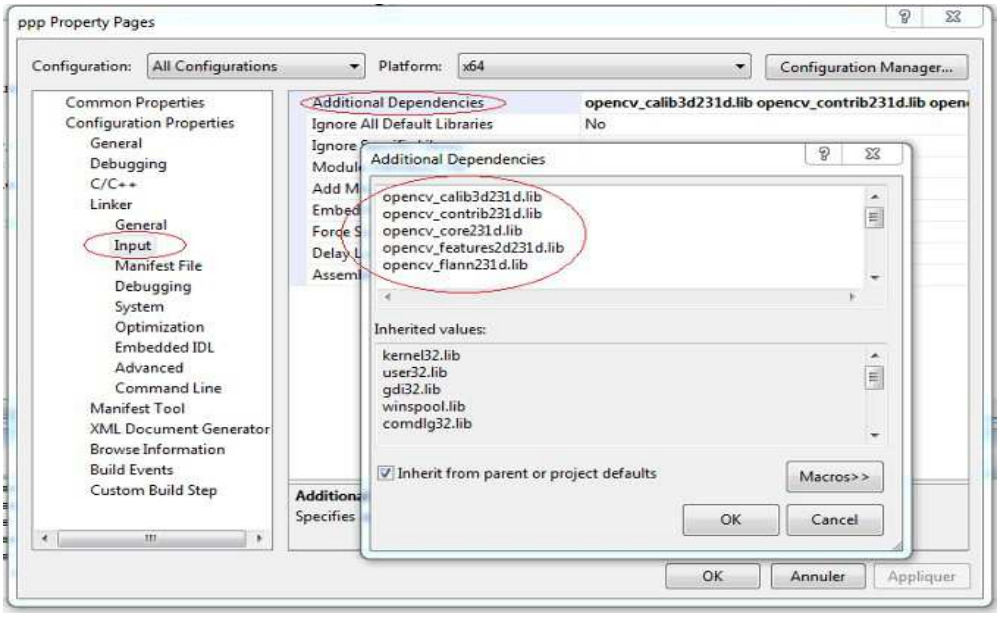


Figure I. 5 : Configuration nécessaire de Microsoft Visual Studio 2008 pour Open-CV

5. Conclusion

Dans ce chapitre, nous avons présentés, une configuration de la bibliothèque visuelle open-cv avec Visual studio 2008.

Open-Cv est un outil d'apprentissage très utilisé dans le domaine de la vision par ordinateur, il est gratuit et open source.
Un des buts d'Open-CV est d'aider les gens a construire rapidement des applications sophistiquées de vision a l'aide d'infrastructure simple de vision par ordinateur.

CHAPITRE II

Généralités sur les images et les Vidéos

1. Introduction

Le traitement des images numériques et des vidéos est une discipline nouvelle qui s‟est développée rapidement grace à l‟émergence des nouvelles technologies de l‟information qui permet de modifier une image ou une vidéo dans le but d'extraire des informations ou d'améliorer qualitativement ou quantitativement avec le codage de la compression des grand volumes d'informations qui s'impose alors comme une étape incontournable pour optimiser la taille mémoire à leurs archivages ainsi que le temps nécessaire à leur transmission à distance via les réseaux informatiques.

2. Définition de l'Image

L'image est une représentation d'une personne ou d'un objet par la peinture, la sculpture, le dessin, la photographie, le film, caméra etc.

C'est aussi un ensemble structuré d'informations qui, après affichage sur l'écran, ont une signification pour l'œil humain. Sa forme peut être analogique (ex: négatif, photographie, vidéo..) ou numérique (images numérisées suivant divers formats (images compressées ou non…) ou obtenues par des capteurs fournissant des images numérisées) et dans ce cas un traitement par ordinateur est possible.

3. Image numérique

Une image numérique est une image dont le support est stocké sous forme binaire dans un fichier informatique. Celle-ci peut être obtenue soit à partir de capteurs optiques (appareil photo, caméra, scanneur…) ou créée à partir de logiciels (Paintbrush, libreoffice…).

Chaque image numérique est constituée d'un nombre donné de lignes. Chaque ligne comporte un nombre de point donnés. L'ensemble constitue une matrice. Ces points sont dénommés pixel (de l'anglais Picture élément et noté souvent px). Chaque « case » de cette matrice contient des nombres caractéristiques à la couleur attribuée au pixel[2][3].

Figure II.1 : des images numériques

4. Caractéristiques d'une image numérique

L'image est un ensemble structuré d'informations caractérisées par les paramètres suivants:

4.1. Pixel

Contraction de l'expression anglaise " Picture éléments ": éléments d'image, le pixel est le plus petit point de l'image, c'est une entité calculable qui peut recevoir une structure et une quantification. Si le bit est la plus petite unité d'information que peut traiter un ordinateur, le pixel est le plus petit élément que peuvent manipuler les matériels et logiciels d'affichage ou d'impression. La lettre A, par exemple, peut être affichée comme un groupe de pixels dans la figure ci-dessous :

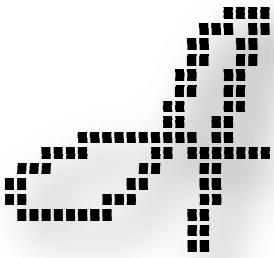


Figure II.2: groupe de pixels représentant la lettre A

Dans le cas d'une image monochrome, chaque pixel est codé sur un octet, et la taille mémoire nécessaire pour afficher une telle image est directement liée à la taille de l'image. Dans une image couleur (R.V.B.), un pixel peut être représenté sur trois octets : un octet pour chacune des couleurs : Rouge (R), Vert (V) et Bleu (B).

4.2. Dimension

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels).
Le nombre de lignes de cette matrice multiplié par le nombre de colonnes nous donne le nombre total de pixels dans une image.

4.3. Résolution

C'est la clarté ou la finesse de détails atteinte par un moniteur ou une imprimante dans la production d'images. Sur les moniteurs d'ordinateurs, la résolution est exprimée en nombre de pixels par unité de mesure (pouce ou centimètre). On utilise aussi le mot résolution pour désigner le nombre total de pixels affichables horizontalement ou verticalement sur un moniteur; plus grand est ce nombre, meilleure est la résolution**.**

4.4. Bruit

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur**.**

Figure II.3 : Image sans bruit. **Figure II.4 :** Image Bruitée.

4.5. Luminance

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. Une bonne luminance se caractérise par :

- ✓ Des images lumineuses (brillantes).
- ✓ Un bon contraste : il faut éviter les images où la gamme de contraste tend vers le blanc ou le noir; ces images entraînent des pertes de détails dans les zones sombres ou lumineuses.
- ✓ L'absence de parasites.

4.6. Histogramme

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image, on modifie souvent l'histogramme correspondant. Il permet de donner un grand nombre d'information sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est repartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée.

Il peut être utilisé pour améliorer la qualité d'une image (Rehaussement d'image) en introduisant quelques modifications, pour pouvoir extraire les informations utiles de celle-ci.

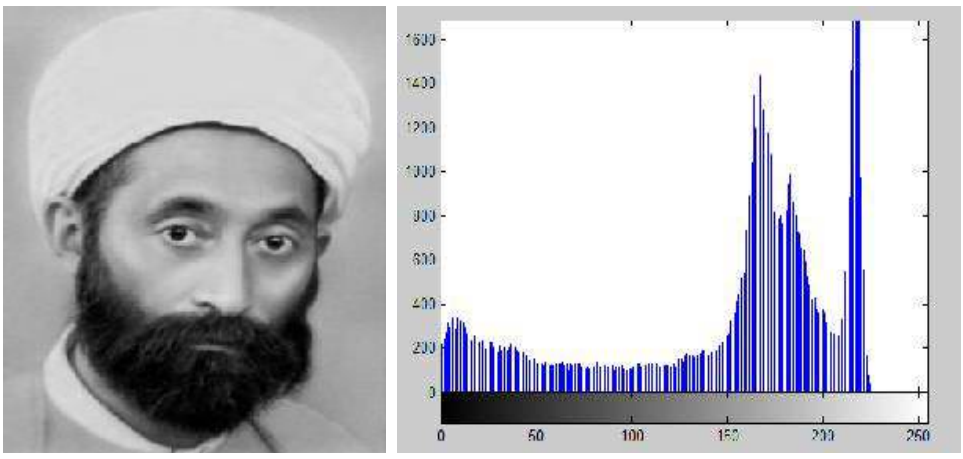


Figure II.5: L'histogramme de l'image

Par fois dans une image naturelle, une majorité de pixels ont une valeur inférieure à la luminance moyenne. Ceci entraîne un problème qui est les détails dans régions sombres

4.7. Contours et Textures

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. Les textures décrivent la structure de ceux-ci. L'extraction de contour consiste à identifier dans l'image les points qui séparent deux textures différentes.

Figure II.6 : Contour d'une image.

4.8. Contraste

C'est l'opposition marquée entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'images.

Si L1 et L2 sont les degrés de luminosité respectivement de deux zones voisines A1 et A2 d'une image, le contraste C est défini par le rapport :

$$C = \frac{L1-L2}{L1+L2} \qquad (1.1)$$

4.9. Le poids de l'image

Le poids d'une image se détermine en fonction de ces deux paramètres: dimensions, profondeur. Le poids de l'image est alors égal à sa dimension multipliée par sa profondeur. Par exemple, pour une image 640x480 en vraies couleurs (True colors) :

- ✓ Nombre de pixels (dimension) : 640 x 480 = 307200.
- ✓ Poids de chaque pixel (profondeur) : 24 bits = 3 octets.
- ✓ Le poids de l'image est ainsi égal à : 307200 x 3 = 921600 octets.

5. Images Bitmap et image vectorielles

Les images appartiennent à deux grandes familles : bitmap (image-bit) ou raster et vectorielle. Alors qu'une image vectorielle est décrite à l'aide des équations mathématiques, une image bitmap est constituée de pixels et se réduit donc à une matrice de points. Si les images vectorielles peuvent être manipulées avec beaucoup de facilité, les modifications de taille, par exemple, apportées à une image bitmap ne sont pas sans incidence. La figure ci dessous montre la différence entre les deux familles.

Figure II.7 : Différence entre l'image vectorielle et l'image matricielle.

6. Codages des couleurs

Usuellement on distingue 3 grands types de couleurs pour une image numérique :

- ✓ Le noir et blanc
- ✓ Les niveaux de gris
- ✓ La couleur.

Ces types sont généralement à choisir lors d'une numérisation par scanner ou lors de la configuration d'un appareil photographique.

Figure II.8: Image originale

6.1. Image noir et blanc

Le noir et blanc est le plus simple. Le contenu de chaque case de la matrice est soit un 0 (noir) soit 1 (blanc). Le nombre de couleurs n'est que de 2 et le rendu de l'image le moins performant mais parfois suffisant dans le cadre par exemple de documents scripturaux.

Figure II.9: Image noire et blanc

La taille du fichier obtenu, avec la même résolution (507x676) que celle de l'image initiale, est de 42,3 ko (au lieu de 980 ko !).

6.2. Niveaux de gris

Le codage dit en niveaux de gris permet d'obtenir plus de nuances que le simple noir et blanc.

Il offre des possibilités supplémentaires pour coder le niveau de l'intensité lumineuse. La couleur est codée souvent sur un octet soit 8 bits ce qui offre la possibilité d'obtenir 256 niveau de gris (0 pour le noir et 255 pour le blanc). On peut aussi le

faire avec16 niveaux de gris(4 bits).

Figure II.10 : Image 'fleur' 256 niveaux de gris gris(taille : 336 ko)

Figure II.11 : Image 'fleur' 16 niveaux de (taille : 170 ko)

De plus pour l'image codée sur 256 niveaux de gris, la valeur du pixel de coordonnées (54,51) (situé en haut à gauche « dans le ciel ») est de 120.

Plus le niveau de gris est élevé, meilleur est la distinction des détails sur l'image. L'usage de ce codage est utilisé fréquemment pour la presse écrite ou l'envoi par messagerie électronique de fichier d'image de taille réduite avec une perte de lisibilité de l'image moindre.

6.3. Image couleur

6.3.1. Principe

La couleur d'un pixel est obtenue, comme le ferait un peintre, par le mélange de couleurs fondamentales. Il ne s'agit pas ici de décrire toutes les techniques utilisées. Nous allons décrire un des principes les plus couramment utilisé qui est celui de la synthèse additive.

6.3.2. Codage RVB

Le principe consiste à mélanger les 3 couleurs : rouge, vert et bleu (noté RVB ou RGB en anglais). A l'aide de ces 3 couleurs, on obtient toute une palette de nuances allant du noir au blanc. A chaque couleur est associé un octet (donc 256 niveaux de luminosité) de chacune des couleurs fondamentales.

Rouge	Vert	Bleu	Couleur
0	0	0	Noir
0	0	1	Nuance de noir
255	0	0	Rouge
0	255	0	Vert
0	0	255	Bleu
128	128	128	Gris
255	255	255	Blanc

Tableau II.1 :Principe codage de la couleur

Un pixel 'couleur' est alors codé avec 3 octets et on a alors la possibilité d'obtenir 224 possibilités de couleurs soit de l'ordre de 16 millions de couleurs différentes.

On dit que les images obtenues sont en couleurs « vraies ». La qualité colorimétrique obtenue est celle d'une photographie argentique couleur.

7. Les formats d'image

7.1. Définition

Lors de son enregistrement une image est stockée suivant un format d'image précis. Ce format doit permettre de stocker l'information de l'image avec un minimum de perte d'informations. Il existe ainsi différents formats qui pourront favoriser soit la conservation de la qualité soit la diminution de la taille du fichier informatique.

Le tableau suivant donne les principales caractéristiques des principaux standards utilisés[7][8].

Format	Type	Compression données	Nombre couleurs	Affichage	Usage
BMP	matriciel	non	de 2 à 16 millions	non	Image non dégradée ; Taille fichier importante.
JPG	Matriciel	oui	16 millions	oui	Taux de compression réglable ; Perte de qualité.
GIF	Matriciel	oui	De 2 à 256 couleurs	oui	Pas de perte de qualité ; Usage pour Internet.
TIFF	Matriciel	oui	16 millions	non	Pas d'usage Internet
PNG	Matriciel	oui	de 2 à 16 millions	Oui	Recommandé pour Internet
SVG	Vectoriel	oui	16 millions	non	Usage cartographie, animations

Tableau II.2 : Principaux formats

8. Définition d'une vidéo numérique

Le mot « vidéo » vient du latin « *video* » qui signifie : « je vois ». C'est l'apocope de vidéophonie ou vidéogramme. Le substantif vidéo s'accorde en nombre, cependant, l'adjectif reste toujours invariable

Une vidéo est une succession d'images à une certaine cadence. L'œil humain a comme caractéristique d'être capable de distinguer environ 20 images par seconde. Ainsi, en affichant plus de 20 images par seconde, il est possible de tromper l'œil et de lui faire croire à une image animée. On caractérise la fluidité d'une vidéo par le nombre d'images par secondes (en anglais frame rate), exprimé en FPS (Frames per second, en

français frames par seconde). D'autre part la vidéo au sens multimédia du terme est généralement accompagnée de son, c'est-à-dire de données audio[6][9].

8.1. Représentation d'une séquence vidéo

Une séquence vidéo brute est une suite d'images fixes, qui peut être caractérisée par trois principaux paramètres :

Résolution en luminance

Détermine le nombre de nuances ou de couleurs possibles pour un pixel. Celle-ci est généralement de 8 bits pour les niveaux de gris et de 24 bits pour les séquences en couleurs.

Résolution spatiale

Définit le nombre de lignes et de colonnes de la matrice de pixels.

Résolution temporelle

Est le nombre d'images par seconde. La valeur de ces trois paramètres détermine l'espace mémoire nécessaire pour stocker séquence vidéo.

Par exemple

Séquence de durée (1sec), résolution de 720 par 576 pixels, un codage des couleurs sur 24 bits, et une fréquence de 25 images par seconde

Débit requis= 31 MO/s (mégabits par seconde).

Le débit d'une séquence vidéo brute est très élevé comparé aux débits et à l'espace offert par les moyens de stockage et de transferts actuels.

8.2. Les différents formats de fichiers vidéo.

Voici la liste des formats de fichiers vidéo les plus utilisés ainsi que leurs principales Caractéristiques.

- ***Le format Quick Time.***

Le **Quick Time** est le format de codage de données vidéo et audio créé par Apple. Il est l'un des plus anciens formats de compression vidéo.

Ce format sert aussi bien à la vidéo qu'au son audio. Son gros avantage est la taille réduite des Fichiers vidéo (rarement supérieur à quelques Mo) mais la qualité d'image en contrepartie est Souvent réduite au minimum.

- ***Le format AVI***

Le format **AVI** est le format de fichier vidéo créé par Microsoft. Sigle de « Audio Vidéo Interleave », son nom d'origine du format est en réalité " Video for Windows ".
Ce format prévoit une définition d'image maximale de 320 x 240 pixels et une cadence maximale de 30 images/seconde, mais en contrepartie, la compression et la décompression des images sont effectuée par logiciel de lecture que l'on utilise.
L'avantage est que les applications multimédias faisant appel au format AVI peuvent toucher une large base d'utilisateurs disposant d'un PC en configuration standard. Cependant les fichiers sont très lourds et demande une configuration matérielle assez performante.

- ***Les formats MPEG***

Le **MPEG** (Motion Picture Experts Group) Format de fichier audio et vidéo utilisant la compression avec perte. Utilisé pour la diffusion en continu sur Internet (les teaming). Le principe est d'obtenir des fichiers légers en supprimant les parties d'images fixes lors du passage d'une image à une autre image [43]. Il existe plusieurs standards de MPEG comme le MPEG-1, le MPEG-2, le MPEG-4, et MPEG-7.

- ***Le format Windows Média File***

Le format **Windows Média File** est le format de codage de données multimédias créé par Microsoft. Les formats audio et vidéo de la technologie de Microsoft, Windows Media, sont présentés comme globalement les plus efficaces aujourd'hui.
Il reste qu'outre la qualité de ses codecs audio et vidéo, WMF est le premier format à assurer également la gestion des droits (identité du détenteur des droits sur la séquence audio ou vidéo), question majeure dans le cadre de la diffusion des contenus multimédias par Internet.

- ***Le format* DivX**

" Divx " (sigle de Digital Video eXpress) est un format de compression numérique de contenus vidéo fondé sur MPEG-4.

- ***Le format Xvid.***

Le Xvid est un format de compression de vidéo entièrement libre de droit. Il concurrence DivX et en reprend d'ailleurs le nom, écrit à l'envers. Il est aujourd'hui de meilleure qualité.

- ***Le format mp4***

Ce format (format vidéo compressé) est très utilisé sur Internet. Il permet l'affichage d'un flux vidéo à mesure qu'il est diffusé. Pour lire une vidéo enregistrée dans le format MP4, il suffit d'utiliser un lecteur multimédia comme le lecteur VLC.

9 . Compression de la vidéo numérique

La compression vidéo est une méthode de compression de données, qui consiste à réduire la quantité de données, en minimisant l'impact sur la qualité visuelle de la vidéo. Cela permet de réduire l'espace de stockage nécessaire et facilite aussi sa diffusion. C'est la lourde tâche du codec qui définira la technique et les paramètres de compression puis de décompression pour la lecture. On distingue les compressions sans perte (au cours de laquelle aucune donnée n'est supprimée de l'image) et les compressions avec perte (qui élimine des données de manière sélective).

9.1. Codec

Pour réaliser la compression/décompression d'un fichier vidéo, l'ordinateur à besoin d'un logiciel appelé **Codec**. Le mot codec provient de la concaténation de deux mots : compresseur et décompresseur, ou encore codeur/décodeur.

Le processus de compression consiste à traiter la source vidéo à l‟aide d‟un algorithme afin de créer un fichier compressé prêt pour la transmission ou le stockage. Pour lire le fichier compressé, un algorithme inverse est appliqué, ce qui permet d‟obtenir une vidéo contenant pratiquement le mrme contenu que la source vidéo d‟origine. Le temps pris pour la compression, l‟envoi, la décompression et l‟affichage d‟un fichier est ce que l‟on appelle la latence. Plus l‟algorithme de compression est perfectionné, plus la latence est élevée.

Les codecs vidéo de différentes normes ne sont normalement pas compatibles les uns avec les autres. En d‟autres termes, le contenu vidéo compressé à l‟aide d‟un algorithme ne peut pas étre décompressé à l‟aide d‟un algorithme différent.

Par exemple, un décodeur MPEG-4 n‟est pas compatible avec un encodeur H.264. L‟explication réside dans le fait qu‟un algorithme ne peut pas décoder correctement le résultat obtenu par un autre algorithme, mais il est possible d‟utiliser plusieurs algorithmes différents dans le même logiciel ou matériel, ce qui autorise la coexistence de plusieurs formats .

9.2. Les lecteurs de vidéo

Pour lire un fichier vidéo sur son ordinateur on se doit d'utiliser en plus des codecs un logiciel de lecture.

Le plus connu est bien entendu « Windows Média Player ».

Ce dernier est fourni avec le système d'exploitation Windows et permet de lire un grand nombre de formats si bien entendu les codecs sont correctement installés.

Cependant certains fichiers ne peuvent être lu qu'avec un lecteur spécifique. Les fichiers « Quick Time » ne peuvent être lu qu'avec le lecteur qui porte le même nom. De même, les fichiers Real Vidéo ne peuvent être lu qu'avec le lecteur « Real Player ».

Le tableau suivant donne les principaux lecteurs de vidéo

Format de la vidéo	Extension du fichier	Codec spécifique à installer en plus du pack fourni sous Windows	Lecteur spécifique à utiliser
Quick Time	.mov	non	Quick Time
AVI	.avi	non	Windows Media Player
MPEG	.mpeg ou .mpg	non	Windows Media Player
Real Vidéo	.ra	non	Real Player
Windows Media	.wmf	Une mise à jour de Windows est à faire	Windows Media Player
DivX	.avi ou .mpg	oui	Windows Media Player
Xvid	.xvid ou .mpg	oui	Windows Media Player

TableauII.3 : Principaux lecteurs de vidéo

9.3. Algorithmes de compression Vidéo

9.3.1. Technique intra-image (compression spatiale)

Les algorithmes de compression vidéo intra-image s'appuient sur les principes classiques de compression d'image où chaque image est travaillée individuellement. Les données sont réduites en supprimant les informations redondantes ou inutiles car non remarquables par l'oeil humain. Il existe plusieurs techniques. En voici quelques exemples, sans rentrer dans les

détails mathématiques.

- **Transformation des couleurs** : le système de stockage colorimétrique est modifié par un système moins gourmand en taille mémoire. Par exemple, le codec MPEG (1,2 ou 4) transforme le classique système RVB (image de gauche) qui enregistre pour chaque pixel , la valeur du rouge, du vert et du bleu, en un système plus proche de la représentation de l'oeil humain et sollicitant moins d'espace de stockage, le système YUV intégrant une composante de luminosité et deux valeurs de chrominance (image de droite).

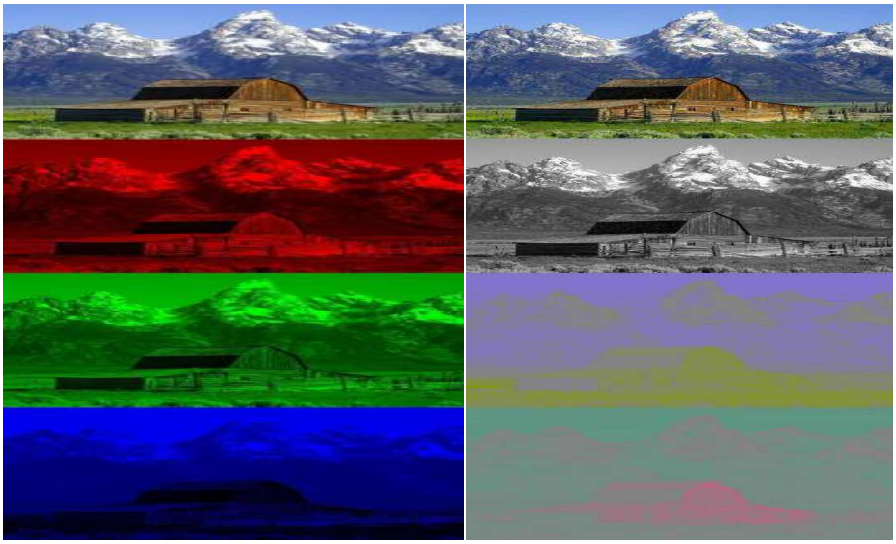


Figure II.12 : Exemple de technique intra-image par transformation des couleurs.

- **Elimination les redondances**

Dans une image, des blocs peuvent se reproduire à l'identique ou, éventuellement, avec de très légères variantes. Le premier bloc d'une série étant codé, il devient redondant de coder les autres blocs qui lui sont identiques, ou qui disposent d'un composant colorimétrique identique. Il suffit d'indiquer qu'il y a répétition du premier bloc. La compression spatiale identifie les zones redondantes au sein d'une même image et ne stocke que les différences. Plusieurs techniques existent et peuvent être combinées pour un meilleur résultat. C'est le cas du format de compression d'image JPEG utilisé par le codec vidéo MJPEG.

Figure II.13 : Exemple de technique intra-image par Elimination les redondances.

9.3.2. Technique inter-image (compression temporelle)

En examinant une vidéo, il est facile de remarquer qu'il existe dans certaines séquences des éléments redondants fixes ou mobiles mais suivant un mouvement logique. A l'échelle du pixel, on parle donc de corrélation temporelle entre des images successives. Dès lors, une image est décrite en fonction de sa différence par rapport à la précédente. Les zones redondantes sont reprises des images précédentes. Cela implique différents mécanismes d'identification des redondances.

- **Le codage différentiel :** si le premier mécanisme, où une image est comparée à une image de référence et seuls les pixels qui ont changé par rapport à cette image de référence sont codés. Cela permet de réduire le nombre de valeurs codées et stockées pour chaque pixel. Le codage différentiel compare des éléments statiques d‟une image à l‟autre .

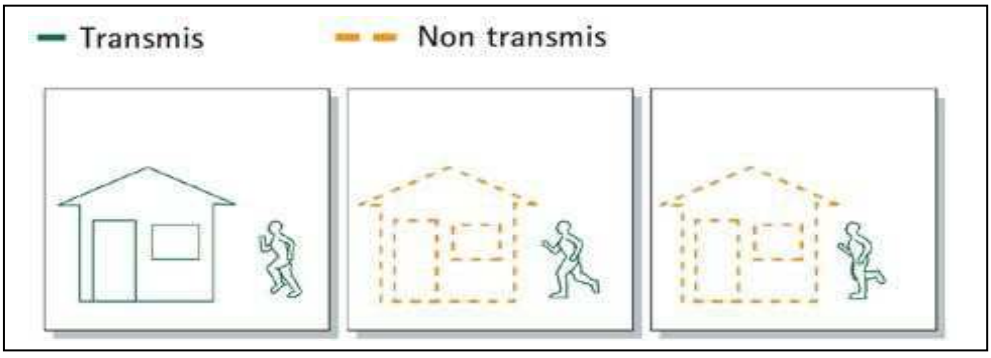


Figure II. 14 : Exemple sur la technique Inter-image par le codage différentiel .

- **La compensation de mouvement**

Cette technique tient compte du fait qu'une bonne partie de ce qui compose une nouvelle image dans une séquence vidéo se retrouve dans une image adjacente, mais pas forcément au même endroit. Il est donc possible de composer ou de « prédire » une nouvelle image bloc par bloc en recherchant un bloc identique dans une image de référence. Si un bloc référent est trouvé, l'encodeur code l'emplacement dans l'image de référence où se situe le bloc identique ainsi que le vecteur de mouvement qui utilise moins d'espace que le codage du contenu réel d'un bloc.

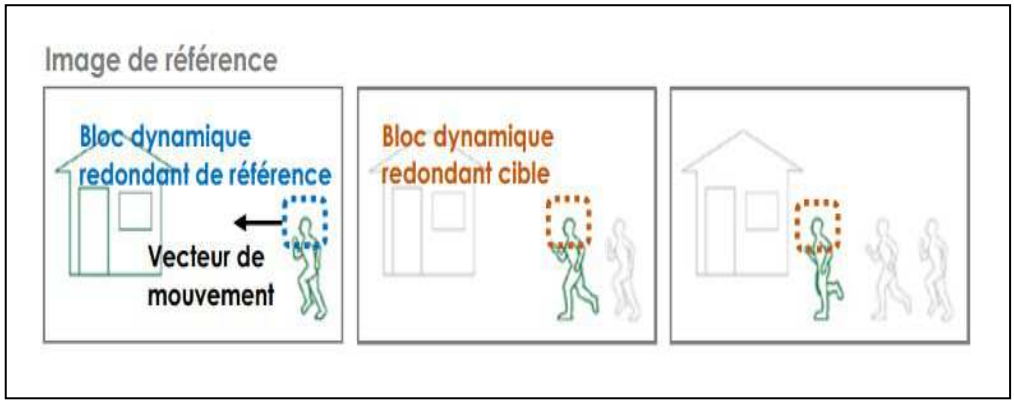


Figure II.15 : Exemple sur la technique Inter-image par La compensation de mouvement

Avec la prédiction inter-images, chaque image d'une séquence est classifiée comme un certain type d'image (image I, image P ou image B).

Une image I : est une image de référence également appelée **I**ntra-frame. C'est une image autonome c'est-à-dire qui peut être décodée sans référence à d'autres images.
La première image d'une séquence vidéo est toujours une image I. Elle fait généralement l'objet d'une compression spatiale intra-image. L'intégration régulière d'image I garantit la qualité résultant de la décompression.

Une image P : est une image **P**rédictive qui fait référence aux parties des images I et/ou P antérieures pour le codage de l'image.
Une image P nécessite généralement moins de bits qu'une image I, mais elle peut être sensible aux erreurs en raison de la dépendance complexe vis-à-vis des images P et/ou I antérieures.

Une image B : est une inter-image **B**i-prédictive qui fait référence à la fois à une image antérieure et à une image future.

Une image devant être compressée avec la méthode B est comparée à toutes les images de type I ou P précédentes ou suivantes, à la recherche de la plus ressemblante et on définit les changements et vecteurs de mouvement.

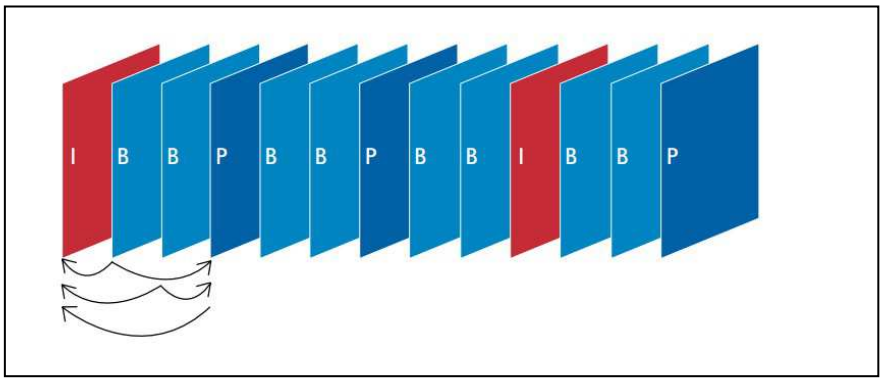


Figure II.16 : Séquence type avec des images I, B et P

10. Conclusion

Dans ce chapitre, nous avons présentés une étude générale sur les caractéristiques fondamentales des images et des vidéos dans le but de les améliorer ou d'extraire des informations pertinents. Donc, les objectifs principale de la compression est de réduire la taille des images et des vidéos.

CHAPITRE III

Les techniques de Traitements des images

1. Introduction

Le traitement d'images et de vidéo est une discipline nouvelle qui s'est développée rapidement grâce à l'émergence des nouvelles technologies de l'information. Il s'appuie notamment sur les mathématiques liées à l'information, le traitement du signal, les systèmes électroniques, et sur l'avancée des capacités de calcul des microprocesseurs développés.
Le traitement d'images et de vidéo est l'ensemble des méthodes et techniques utilisées pour améliorer le contenu et la qualité d'une image ou vidéo ou d'en extraire de l'information.

2. Nécessité de traiter les images numériques

L'acquisition d'images numériques (sous forme de matrices de nombres) transforme un signal continu analogique (une radiation lumineuse) en échantillonnage numérique discontinu. Il y a donc des pertes d'information dues aux méthodes d'échantillonnage et d'autres dues aux erreurs d'acquisition (capteur défectueux, couverture nuageuse, défauts d'optique, de parallaxe, etc.).

Le premier problème est pris en compte en respectant le principe d'échantillonnage énoncé plus haut, la fréquence d'acquisition doit être double de la fréquence maximale du signal.

Le deuxième problème est inhérent à la technologie des capteurs et à des facteurs qui échappent en partie au contrôle de l'utilisateur. Ce n'est que par une manipulation des données enregistrées qu'on peut y remédier. D'où les traitements d'images et les filtrages [9][3].

3. Les étapes du traitement artificiel d'une scène visuelle

3.1 Acquisition des données images

L'acquisition d'images constitue un des maillons essentiels de toute chaîne de conception et de production d'images. Pour pouvoir manipuler une image sur un système informatique, il est avant tout nécessaire de lui faire subir une transformation qui la rendra lisible et manipulable par ce système. Le passage de cet objet externe (l'image d'origine) à sa représentation interne (dans l'unité de traitement) se fait grâce à une procédure de

numérisation. Ces systèmes de saisie, dénommés optiques, peuvent être classés en deux catégories principales : les caméras numériques et les scanners.

Le développement technologique a permis l'apparition de nouveaux périphériques d'acquisition appelés cartes d'acquisition, qui fonctionnent à l'instar des caméras vidéo, grâce à un capteur C.C.D. (Charge Coupled Device). La carte d'acquisition reçoit les images de la camera, de la T.V. ou du scanner afin de les convertir en informations binaires qui seront stockées dans un fichier[16].

3.2. Pré-traitement

Le pré-traitement regroupe toutes les opérations de manipulation de l'image qui permettent d'en améliorer la qualité. Ces manipulations produisent une nouvelle image. On y retrouve différentes techniques :

- La compression : Réduction du volume de l'image.
- La restauration : Correction des défauts dus à une source de dégradation.
- L'amélioration : Modification de l'image dont le but de la rendre plus agréable à l'œil.

Le prétraitement d'une image a pour but d'améliorer la qualité des images (LUT , Filtrage).Le but de cette partie est de mettre en œuvre des outils permettant d'effectuer des Traitements et prétraitements sur l'image affichée précédemment avant d'effectuer la phase de décision.

3.3. Analyse

L'analyse est une suite opérations permettant l'extraction de l'information essentielle contenue dans l'image. Elle est essentiellement composée par la phase de segmentation.

A partir de cette segmentation, des techniques de description permettent d'obtenir la description structurelle de l'image.

3.4. Interprétation

L'interprétation a pour but le passage de la description structurelle à la description sémantique en regard à certain objectif. Cet objectif peut être très simple (mesure de certains paramètres

sur des formes) ou beaucoup plus complexe (description du contenu de la scène en termes de concepts non mathématiques).

4. Traitement numérique des images et les vidéos

On peut citer les traitements suivants :

4.1. L'histogramme

L'histogramme est défini comme une fonction discrète qui associe à chaque valeur d'intensité le nombre de pixels prenant cette valeur. La détermination de l'histogramme est donc réalisée en comptant le nombre de pixel pour chaque intensité de l'image. On effectue parfois une quantification, qui regroupe plusieurs valeurs d'intensité en une seule classe, ce qui peut permettre de mieux visualiser la distribution des intensités de l'image.

Les histogrammes sont en général normalisés, en divisant les valeurs de chaque classe par le nombre total de pixels de l'image. La valeur d'une classe varie alors entre 0 et 1, et peut s'interpréter comme la probabilité d'occurrence de la classe dans l'image[15][16].

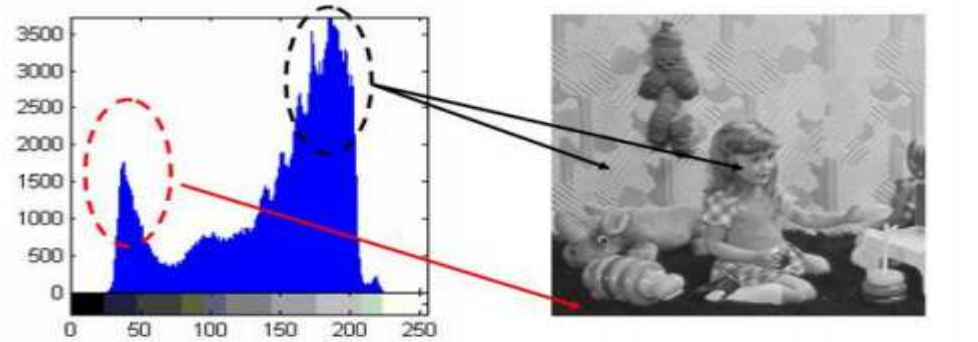


Figure III. 1: Histogramme obtenu pour l'image

4.2. Filtrage D'images

Pour améliorer la qualité visuelle de l'image, on doit éliminer les effets des bruits (parasites) en lui faisant subir un traitement appelé filtrage.

4.2.1 .Les filtres de convolution

Beaucoup de traitements d'images sont basés sur les produits de convolutions, dont nous allons expliquer le principe.

L'image numérique étant en quelque sorte une carte de pixels, on peut identifier chaque pixel par ses coordonnées X et Y et lui affecter une valeur liée à sa luminosité. On peut utiliser dans le cadre des images numériques une sorte un tableau de X colonnes et Y lignes qui réserve une place pour ranger la valeur de chaque pixel de l'image. En mathématique ce genre de Tableau s'appelle une matrice, et les mathématiciens disposent d'outils pour effectuer des calculs sur les matrices, comme additionner deux matrices, les multiplier, etc ...

Un produit de convolution, est un opérateur mathématique qu'on utilise pour multiplier des matrices entre elles. Dans le cas qui nous intéresse, nous mettons en jeu deux matrices très différentes: la matrice image, très grande (par exemple 512 x 512, ce qui représente 262144 pixels) et une matrice plus petite qu'on appelle le noyau par ce que c'est le "cœur" de tous les changements qui vont affecter l'image.

Le noyau va donc agir sur chacun des pixels, c'est à dire sur chacun des éléments de la matrice "image".

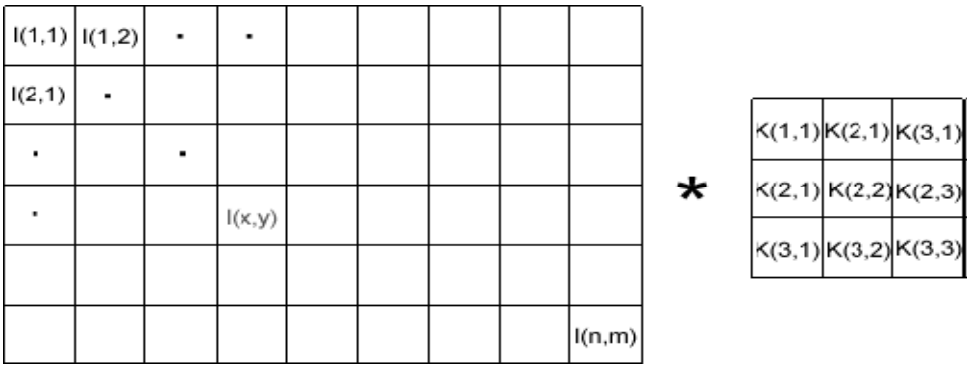


Figure III. 2 : L'image (i) et son noyau (k)

Dans la figure ci-dessus, l'image est représentée par la matrice [i] composée de n x m éléments. Le noyau est quant à lui composé de la matrice carrée [k] de 3x3 éléments.

Appliquer un filtre de convolution consiste à multiplier chacun des pixels de la matrice [i] par le noyau [k]; Pour calculer la valeur d'un pixel I(x, y) de la matrice image, on multiplie sa valeur par celle du pixel central du noyau K(2,2) et on additionne ensuite la valeur des produits des pixels adjacents. Il reste ensuite à diviser le résultat par le nombre d'éléments du noyau, cette dernière opération n'appartient pas au produit de convolution proprement dit, mais elle est nécessaire pour maintenir la dynamique de l'image (différence entre le niveau du pixel le plus élevé et le plus faible) ainsi que sa linéarité.

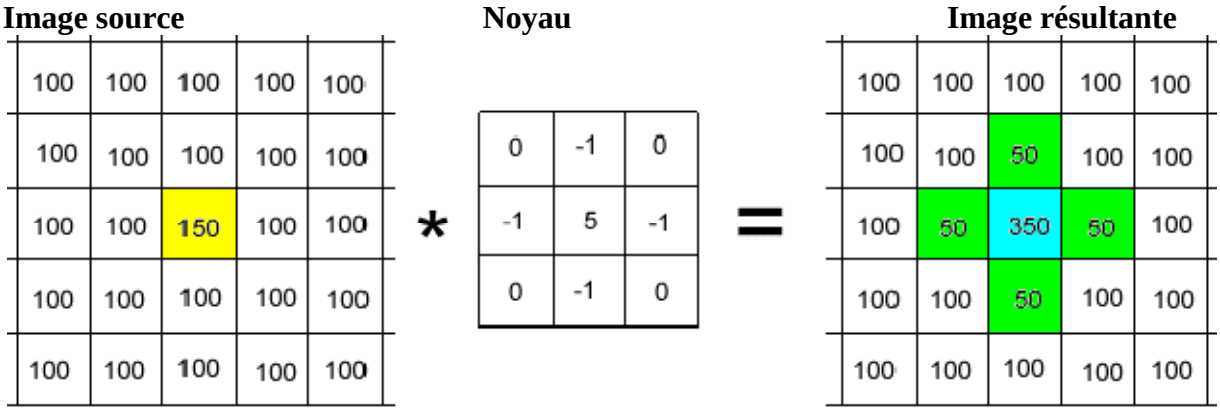


Figure III. 3 : Filtres de convolution

Dans l'exemple ci .dessus, tous les pixels ont une intensité de 100 ADU (ADU est Analog to Digital Unit qui correspond aux unités employées pour donner le résultat de la conversion effectuée lors de la numérisation par le convertisseur analogique vers numérique), sauf le pixel central qui a une intensité de 150. Si on applique un produit de convolution de cet exemple au pixel central on obtient, d'après les règles mathématiques (150×5) + (-1×100) + (-1×100) + (-1×100) + (-1×100) Les pixels des quatre coins n'apparaissent pas parce que leur valeur est zéro. Cet exemple peut correspondre à un fond de ciel d'intensité constante avec une étoile faible dont toute la luminosité est concentrée dans un seul pixel.

Comment souligner la présence de cette étoile ? La réponse est d'augmenter le contraste, mais un problème est de définir exactement ce qu'on appelle le contraste pour évaluer les modifications apportées à l'image

4.2.2. Contraste = I / N

Dans cette relation, I est l'intensité en ADU et N est l'intensité moyenne des pixels adjacents. Dans notre exemple l'intensité du pixel central est I =(150-100) ADU = 50 ADUs (nous soustrayons la valeur du fond de ciel)

Alors que la valeur du fond de ciel est toujours de 100 ADUs, nous avons donc d'après la définition dans l'image de départ un contraste égal à I / N =50 / 100 =0,5 ADU.

4.2. 3. Le filtre passe haut

L'application principale des produits de convolution est la création des filtres « passe haut » et « passe bas ». Un filtre « passe haut » favorise les hautes fréquences spatiales, comme les détails, et de ce fait, il améliore le contraste. Un filtre « passe haut » est caractérisé par un noyau comportant des valeurs négatives autour du pixel central, comme dans l'exemple ci dessous:

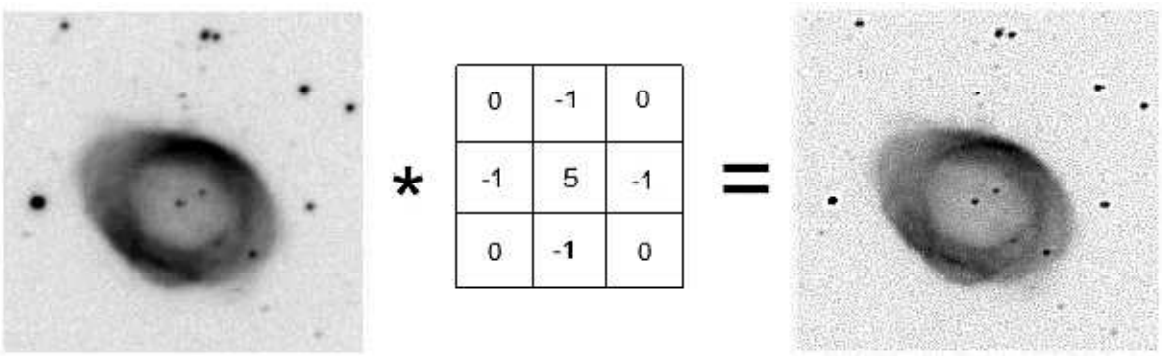


Figure III. 4: Filtrage d'image (passe haut).

4.2.4. Le filtre passe-bas

Les filtres « passe bas » agissent en sens inverse des filtres « passe haut » et Le résultat est, un adoucissement des détails, ainsi qu'une réduction du bruit granuleux.

Niveau. Définit une limite maximum (en ADU) de niveau de pixel pour L'application du filtre. Par exemple le filtre passe-bas peut être appliqué dans la zone sombre du fond de ciel, pour réduire le bruit granuleux sans perdre les détails.

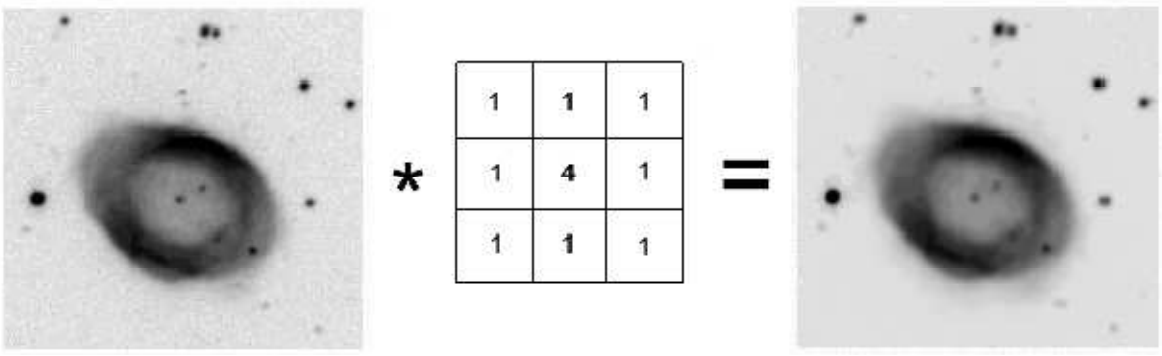


Figure III. 5: Filtrage d'image (passe bas).

4.2.5. Le filtre moyenne

C'est un cas particulier de filtre de convolution « passe-bas », qui remplace chaque pixel par la moyenne des valeurs des pixels adjacents et du pixel central.

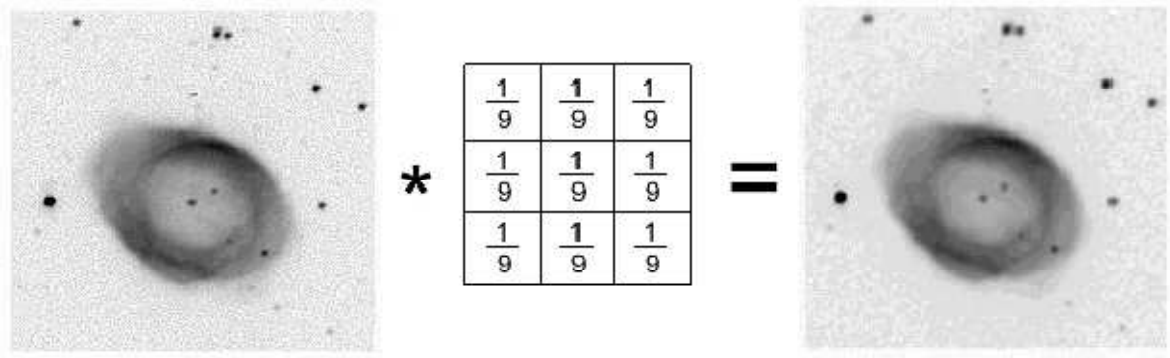


Figure III. 6: Filtrage d'image (moyenne).

Dans l'exemple on voit un 3x3. Utiliser un noyau plus gros comme un 5x5 ou plus grand Encore se révèle très lourd et peut créer une apparence artificielle.

4.2.6. Le filtre médian

Le filtre médian est utilisé pour atténuer des pixels isolés, d'une valeur très différente de leur entourage. C'est le cas par exemple des pixels thermiques chauds ou froids qui ont une intensité très différente de leurs voisins, mais il y d'autres applications, comme par exemple se débarrasser des impacts de rayons cosmiques qui en tombant sur le capteur CCD pendant une exposition y créent un point ou une petite trace rectiligne. Ce point est particulièrement important pour les images prises en haute altitude ou en orbite comme celles du télescope spatial Hubble.

Le filtre médian n'est pas à proprement parler un produit de convolution, mais sa mise en œuvre sur l'image est assez similaire puisqu'un noyau est appliqué sur l'image et collecte les valeurs des pixels. Sur l'exemple ci-dessous le noyau est un 3x3 = 9 éléments. Les neufs éléments extraits de l'image sont ensuite triés dans l'ordre croissant, le résultat ci-dessous est obtenu avec le pixel sur-ligné comme pixel central: 120, 412, 420, 430, 432, 434, 435, 437, 440.

La valeur médiane d'une série est par définition celle qui sépare l'échantillon en deux parties de population égale, ici on voit que c'est la valeur (432) facilement repérable à cause du tri : l'algorithme va donc remplacer la valeur originale par la valeur médiane qui vaut 432 ADU.

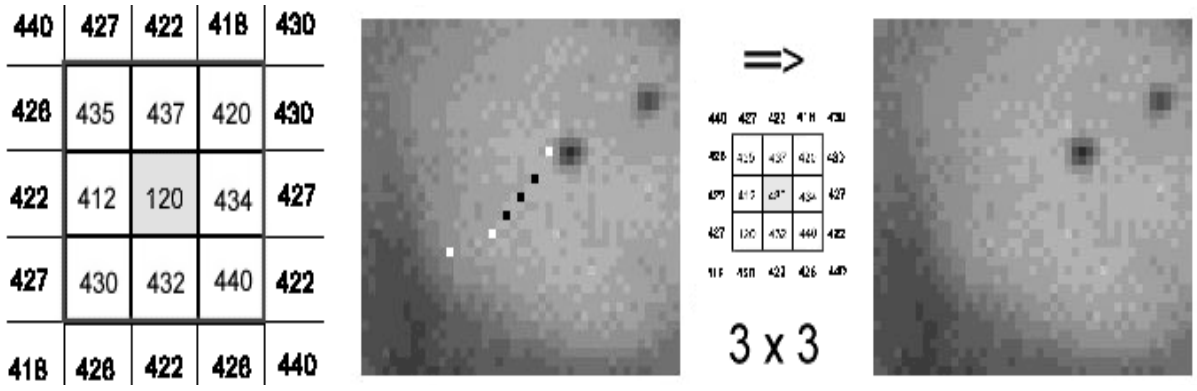


Figure III. 7: Filtrage d'image (median).

Le résultat obtenu avec ce filtre est l'élimination des pixels isolés, en évitant de créer un flou trop important dans l'image. Par contre, à cause de ses propriétés non-linéaires il peut ne pas préserver les qualités photométriques et astrométriques de l'image originale.

4.2.7 .Les filtres de prewitt, sobel, freeman, et kirsch

Dans le même but : détecter avec la plus grande précision les contours naturels "cachés" dans une image. A l'origine ils ont été développés dans le cadre des appareils de vision nocturne, mais ils sont aussi utiles dans l'étude morphologique des objets astronomiques, comme les galaxies spirales.

Le filtre de Sobel utilise par exemple deux noyaux 3x3, l'un pour l'axe horizontal (X) et l'autre pour l'axe vertical (Y) Chacun des noyaux est en fait un filtre gradient, qui sont tous les deux combinés pour créer l'image finale.

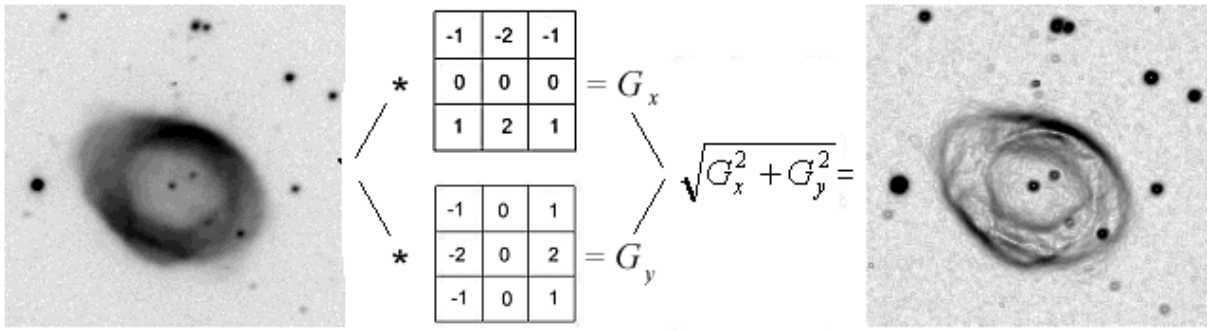


Figure III. 8: Filtrage d'image (SOBEL).

5. La détection des contours

Les contours sont un aspect important de l'analyse de forme et de la reconnaissance de forme. En effet, si je vous demande de m'identifier les formes triangulaires dans un ensemble de figures géométriques, vous parviendrez en quelques secondes an identifier les triangles dans la scène. Le cerveau a identifie ce qui constituait le pour tour de la forme pour isoler les cotes et ainsi compter le nombre d'arrêtes de la figure. C'est donc en analysant le contours des objets que l'on parvient an identifier les formes.

Open CV possède plusieurs fonctions et classes qui permettent d'isoler et d'identifier les contours. Nous verrons dans le prochain chapitre.

5.1. Renforcement des contours par la méthode de Sobel, Laplace et Canny

5.1.1. Filtre de Sobel

Le filtre de Sobel est un operateur utilise en traitement d'image pour la détection de contours. Il s'agit d'un des operateurs les plus simples qui donne toutefois des résultats corrects.

Pour faire simple, l'operateur calcule le gradient de l'intensité de chaque pixel. Ceci indique la Direction de la plus forte variation du clair au sombre, ainsi que le taux de changement dans cette direction. On connait alors les points de changement soudain de luminosité, correspondant probablement à des bords, ainsi que l'orientation de ces bords.

En termes mathématiques, le gradient d'une fonction de deux variables (ici l'intensité en fonction des coordonnées de l'image) est un vecteur de dimension 2 dont les coordonnées sont Les dérivées selon les directions horizontale et verticale. En chaque point, le gradient pointe dans la direction du plus fort changement d'intensité, et sa longueur représente le taux de variation dans cette direction. Le gradient dans une zone d'intensité constante est donc nul. Au niveau d'un contour, le gradient traverse le contour, des intensités les plus sombres aux intensités les plus claires.

Voici les matrices :

$$\begin{matrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{matrix}$$

Matrice de Sobel pour les contours horizontaux

$$\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

Matrice de Sobel pour contours verticaux

Image originale

Image après Sobel horizontal

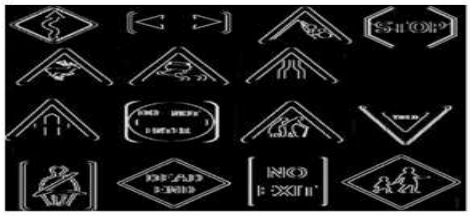


Image après Sobel vertical

Figure III-9: détection de contours (sobel)

5.1.2. Le filtre de Laplace

Le filtre Laplacien est un filtre de convolution particulier utilise pour mettre en valeur les détails qui ont une variation rapide de luminosité.

Il est donc idéal pour rendre visible les contours des objets. D'un point de vue mathématique, le Laplacien est une dérivée d'ordre 2, a deux dimensions et se note $\Delta I(x,y)$ ou $\nabla 2I(x,y)$.

Dans le cas du traitement d'image, l'image de départ f(i,j) n'est pas une fonction continue, mais une fonction discrète a cause de la numérisation effectuée. Mais on peut tout de même obtenir la dérivée seconde (soit le laplacien) avec une bonne approximation grâce aux noyaux de convolution suivants (entre autres).

Matrice de Laplace :

$$\begin{matrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{matrix}$$

Image originale l’image après laplace

Figure III-10: détection de contour (laplace)

5.1.3. Le fitre de canny

Canny a proposé un filtre déterminé analytiquement à partir de trois Critères :

1. Une bonne détection : l'opérateur donne une réponse au voisinage d'un contour ;
2. Une bonne localisation : optimisation de la précision avec laquelle le contour est détecté;
3. Unicité de la réponse : le contour doit provoquer une réponse unique de l'opérateur.

Image original l’image après canny

Figure III.11 : détection de contour (CANNY)

6. La Segmentation

La segmentation est un domaine d'analyse d'image qui consiste à diviser une image en différentes régions homogènes connectées selon certaines caractéristiques probabiliste. Il s'agit principalement de créer des partitions dans l'image en identifiant des pixels similaires. Le critère de similarité permet de contrôler l'aspect final de la segmentation (regroupement par couleur, par homogénéité, par taille).La segmentation a pour objectif de différencier des zones d'intérêt (par exemple de séparer les objets du fond).

6.1. Les différentes approches de segmentation

À ce jour, il existe de nombreuses méthodes de segmentation, que l'on peut regrouper en quatre principales classes :

6.1.1. La segmentation par régions

Les méthodes de segmentation en régions homogènes consistent à trouver des ensembles de pixels qui partagent des propriétés similaires. Les régions homogènes sont construites à partir des parties connexes de ses ensembles. On distingue deux grandes techniques de segmentation en régions : la croissance par région et la décomposition-fusion

6.1.2. Segmentation par croissance de régions

Les méthodes d'accroissement de région sont des méthodes ascendantes consistant à regrouper les pixels vérifiant certaines propriétés d'homogénéité. Le principe des algorithmes par croissance de régions part d'un premier ensemble de régions, qui peuvent être calculées automatiquement (par exemple, les minima de l'image) ou fournies par un utilisateur de manière interactive. Les régions grandissent ensuite par incorporation des pixels les plus similaires suivant un critère donné, jusqu'à ce que toute l'image soit couverte.

6 .1.3 Segmentation par approche contours

La deuxième catégorie s'intéresse aux contours des objets dans l'image. Elle consiste à détecter les points frontières entre 2 zones homogènes de caractéristiques différentes.

Elle nous donne accès aux différentes zones de l'image que l'on pourra après coup (par exemple en remplissant ces régions) isoler pour les traite. La plupart de ces algorithmes sont locaux. Le résultat est en général difficile à exploiter sauf pour des images très contrastées.

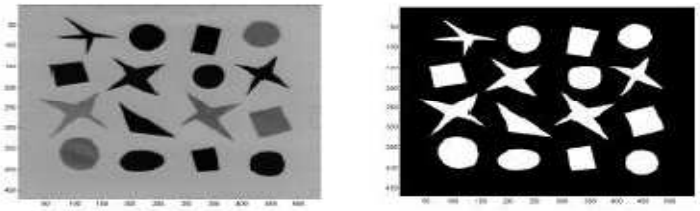
6.1.4 Segmentation par classification ou seuillage

Le seuillage est une méthode simple et très populaire pour la segmentation d'objets dans les images numériques. Il peut être de nature :

- *Globale* : un seuil pour toute l'image;
- *Locale* : un seuil pour une portion de l'image;
- *Adaptative* : un seuil qui s'ajuste selon les images/parties de l'image.

On part ici d'un rapport qu'entretient chaque pixel individuellement avec des informations calculées sur toute l'image, comme par exemple la moyenne des niveaux de gris de l'ensemble des pixels, ou la médiane, permettant de construire *n* classes d'intensité. Lorsque les classes sont déterminées par le choix d'un seuil, on parle de seuillage.

Les pixels appartenant à une même classe et étant connexes forment des régions. Le seuillage constitue une approche simple à la segmentation car plusieurs zones déconnectées ont la même valeur d'étiquette. Le seuillage sert de composante de base à des algorithmes plus complexes de segmentation.



1. Image originale. **2.** Image seuillée.

Figure III.12 : segmentation par seuillage.

Avec ce procédé, le travail le plus important apparaît au moment du choix d'un seuil pertinent pour séparer correctement les différents objets contenus dans l'image. Pour cela, le meilleur moyen est d'analyser l'histogramme des niveaux de gris ou des couleurs de l'image, car il nous permet facilement de détecter les niveaux d'intensité lumineuse qui sont plus abondants que les autres.

6.2. Manipulation de la segmentation

L'utilisation des algorithmes de segmentation seule, ne donne pas toujours des résultats compréhensibles pour leurs utilisations directes, il faut parfois faire subir aux résultats quelques traitements pour que ceux-ci soient plus aisément manipulables

6.2.1 La méthode de recherche des composantes connexes

C'est pourquoi on utilise un algorithme de recherche des composantes connexes, qui parcourt un ensemble de régions faisant partie d'une seule et même entité afin d'étiqueter toutes les régions connexes avec un identificateur unique, et ainsi permettre leur manipulation séparée.

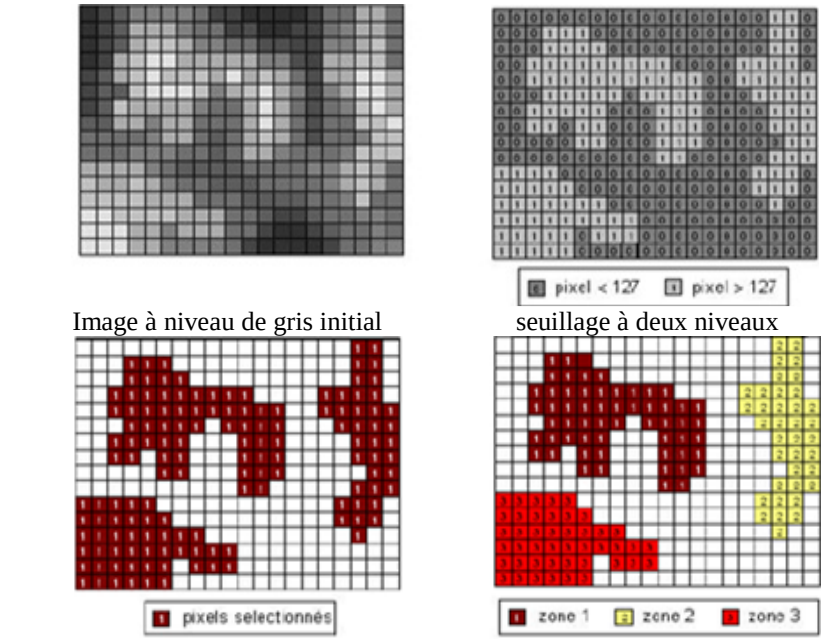


Image à niveau de gris initial — seuillage à deux niveaux

régions sélectionnées pour recherche des composantes connexes — composantes connexes

Figure. III.13: recherche des composantes connexes.

6.2.2 La méthode de suppression des petites régions

Pour réduire le nombre de régions et ainsi diminuer le temps de calcul, on va effectuer une opération de suppression des petites régions.

La méthode mise en œuvre est assez simple: elle consiste, si une région a une taille inférieure à celle requise, de rechercher tous ses voisins. On va alors prendre la plus grande zone adjacente et la fusionner avec la région actuelle.

Mais on va éviter au maximum de fusionner des régions qui sont trop différentes, pour cela on va tout d'abord rechercher parmi les régions mitoyennes les plus proches. Ce qu' on entend par là dépend de l'algorithme utilisé.

7. Conclusion

Le traitement d'image est une phase très important pour amélioration de la qualité de l'image et vidéo.

Dans ce chapitre, nous avons présentés les différentes techniques et les méthodes de traitement des images comme le filtrage, la détection de contour et la segmentation.

Dans le prochain chapitre, nous présentons l'implémentation des différents algorithmes de traitement d'image en utilisant la bibliothèque OpenCV.

Chapitre IV

Implémentation et Résultats

1. Introduction

L'objectif de notre projet est basé sur le développement d'une application de traitement des images et des vidéos en utilisant la bibliothèque OpenCV avec l'utilisation de Visual studio.

2. Environnement du travail

Dans cette section, nous présentons les environnements matériels et logiciels de notre travail.

2.1. Environnement matériel ou le hardware:

Un ordinateur Lenovo avec les caractéristiques suivantes:

Processeur: Intel (R) Core Duo avec CPU 2.8GHz

RAM: 4.00 GO.

Disque Dur: 500 GO.

Webcam intégrée de marque Lenovo USB 2.0 Camera.

2.2. Environnement immatériel ou le software :

Système d'exploitation : Microsoft Windows 7 - 64 bits

Logiciel : Microsoft Visual Studio version 2008

La bibliothèque : OPENCV version 2.3.1

3. Les principes de traitement d'images et de vidéo

3.1. Les opérations pour lire les images ou les vidéos

❖ **Pour les images**

Voici le code source qui est utilisé pour charger une image stockée dans l'ordinateur

```
Mat gray=imread("Dome.jpg",0);
namedWindow( "Gray", 1 );
imshow( "Gray", gray );
```

- ✓ `"Dome.jpg"` **:** nom du fichier à lire.
- ✓ `0:` spécifie le type de couleur.
- ✓ `Gray:` nom de la fenêtre.

- ✓ `gray:` Image à visualiser

Imshow (nom fenêtre, image) : affichage dans l'interface du résultat.

❖ **Pour les vidéos**

En ce qui suit, les codes source utilisés pour la lecture de la vidéo

// lecture de vidéo

```
Mat frame
VideoCapture cam(0);
char c;

    while(1)
    {
        cam >> frame;
```

OU

```
{
    VideoCapture cap(0); //capture the video from web cam

    if ( !cap.isOpened() )  // if not success, exit program
    {
         cout << "Cannot open the web cam" << endl;
         return -1;
    }
```

3.2. L'histogramme d'image

- ✓ **calcHist** () : Cette fonction permet d'obtenir l'histogramme d'une image. Le nombre de paramètres est très Important et cette fonction peut devenir très vite complexe.

- ✓ **Normalize** () : c'est la fonction qui fait la normalisation d'histogramme d'une image.

- ✓ **waitKey () :** Attends que l'utilisateur appuie sur une touche du clavier (retourne le code ASCII).

Le code source de l'histogramme

```
int main(int, char**)
{
    Mat gray=imread("Dome.jpg",0);
    namedWindow( "Gray", 1 );    imshow( "Gray", gray );

    // Initialisation des  parametres
    int histSize = 256;
    float range[] = { 0, 255 };
    const float *ranges[] = { range };

    // Calculate histogram
    MatND hist;// matrice stockant les données (éventuellement
CvSparseMat)

   calcHist( &gray, 1, 0, Mat(), hist, 1, &histSize, ranges, true,
false );


    double total;
    total = gray.rows * gray.cols;
    for( int h = 0; h < histSize; h++ )
         {
            float binVal = hist.at<float>(h);
            cout<<" "<<binVal;
         }

    // affichage de   histogram
    int hist_w = 512; int hist_h = 400;
    int bin_w = cvRound( (double) hist_w/histSize );

    Mat histImage( hist_h, hist_w, CV_8UC1, Scalar( 0,0,0) );
    normalize(hist, hist, 0, histImage.rows, NORM_MINMAX, -1, Mat() );

    for( int i = 1; i < histSize; i++ )
    {
      line( histImage, Point( bin_w*(i-1), hist_h -
cvRound(hist.at<float>(i-1)) ) ,
                       Point( bin_w*(i), hist_h -
cvRound(hist.at<float>(i)) ),
                       Scalar( 255, 0, 0), 2, 8, 0  );
    }

    namedWindow( "Result", 1 );    imshow( "Result", histImage );

    waitKey(0);
    return 0;
}
```

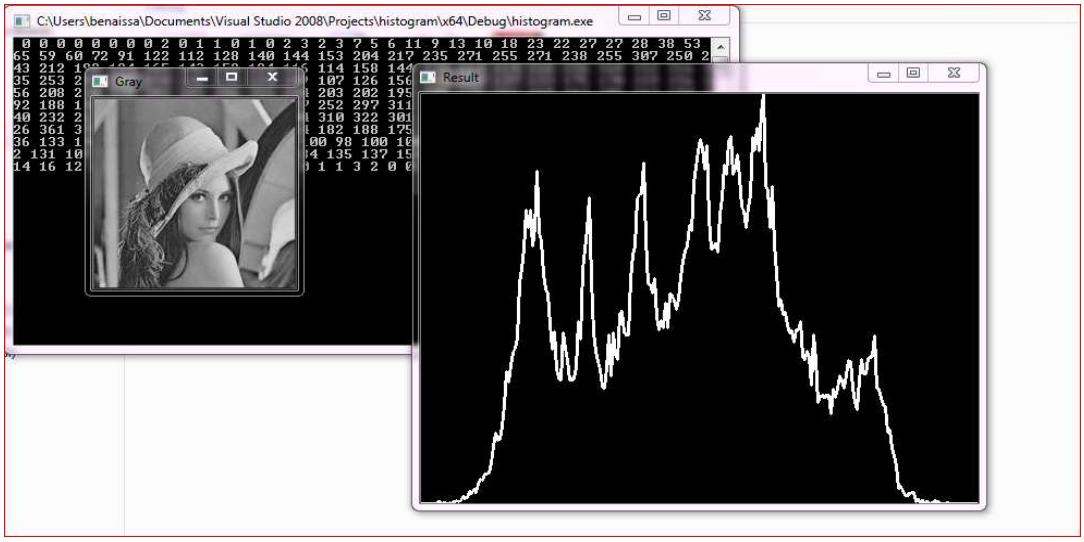


Figure IV. 1 : l'histogramme d'une image.

3.3. Détection de contour d'image

- CvtColor [src, dst, convert] : cette fonction permet la conversion entre deux espaces colorimétriques

✓ src : Image source.

✓ dst : Image de destination.

✓ Convert : code de conversion utiliser les constantes CVBGR2HSV, CVHSV2BGR, etc.

- CVCanny [image, edges, threshold1, threshold2, apertureSize] : cette fonction détecter les bords par la méthode de canny.

✓ Image : image source a une couche.

✓ edges : image de destination, image binaire comportant les bords.

✓ threshld1 : seuil bas.

✓ threshld2 : seuil haut.

✓ aperturesize (taille du noyau de convolution (1 ,3 ,5ou 7)

Le code source pour la détection de contour selon l'algorithme cany

```
int main( )
{
    Mat src1;
    src1 = imread("lena.jpg", CV_LOAD_IMAGE_COLOR);
    namedWindow( "image originale", CV_WINDOW_AUTOSIZE );
    imshow( "image originale", src1 );

    Mat gray, edge, draw;
    cvtColor(src1, gray, CV_BGR2GRAY);

    cvCanny( gray, edge, 50, 150, 3);

    edge.convertTo(draw, CV_8U);
    namedWindow("image contour", CV_WINDOW_AUTOSIZE);
    imshow("image contour", draw);

    waitKey(0);
    return 0;

        }
```

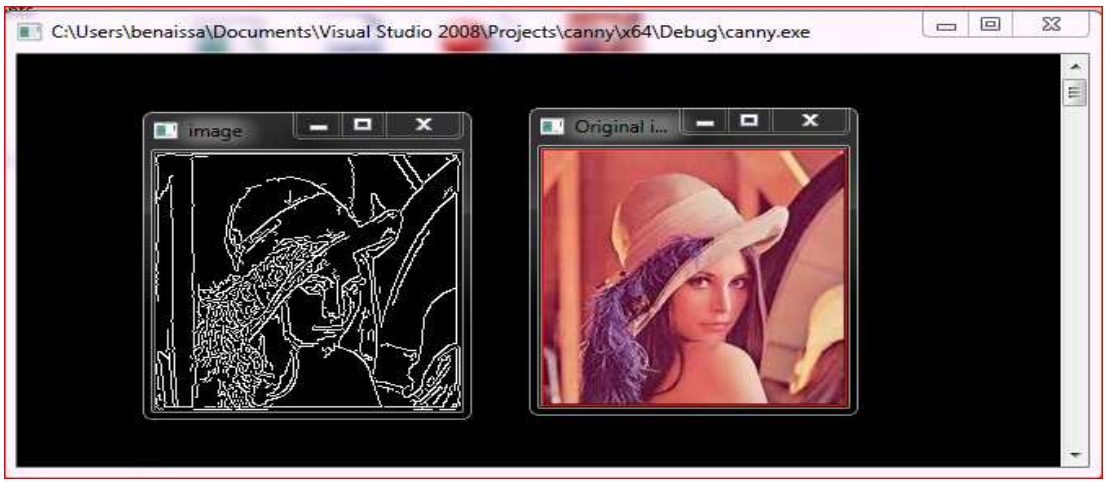


Figure IV. 2 : détection de contour d'une image (canny).

❖ **Pour les vidéos**

Permet les fonctios utilisées

Utilisation de la webcam avec la classe Mat

VideoCapture **:VideoCapture (**int device**)**

Cette fonction est le constructeur dans la classe VideoCapture. Ce constructeur ouvre la Webcam indexée par l'argument et initialise l'objet VideoCapture.

Le code source pour un capture d'un vidéo par le webcam

```
using namespace std;
using namespace cv;

int main(int argc, char *argv[])
{
      // lire le stream vidéo

    Mat frame, img_contours;
    VideoCapture cam(0);     // ouvrir le webcam no. 0
    char c;

    while(1) //boucle infinie
    {
        cam >> frame;
        cvtColor(frame, img_contours, CV_BGR2GRAY); // conversion en
niveau de gris
        Canny(img_contours, img_contours, 127, 127);
        namedWindow("contours");
        imshow("contours", img_contours);
        c = waitKey(20);
        if(c == 27)
            break;
    }

    return 0;
                    }
```

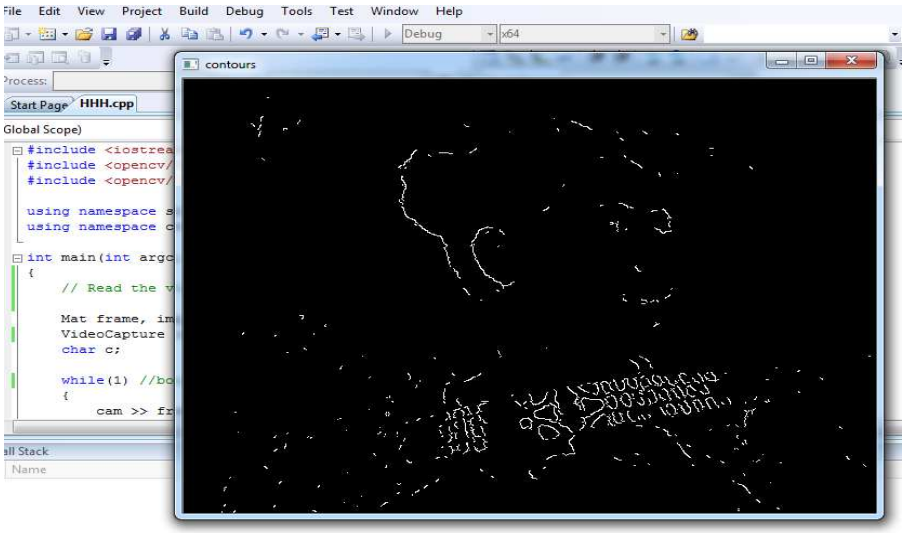


Figure IV. 3 : détection de contour dans les vidéos (canny).

3.4. Filtrage d'image

- **Filtre de sobel**

CVMinMaxLoc[arr] :Obtenir les valeurs mimimales et maximales, ainsi que leur position, d'un CvArr à une couche et à deux dimensions.

- ✓ arr (Integer) : référence d'un CvArr (CvMat ou IplImage).
- ✓ m (Real List) : valeurs min/max et positions sous la forme (min,max,minX,minY,maxX,maxY) .

CV_LOAD_IMAGE_COLOR : L'image est chargée en couleur

WINDOW_AUTOSIZE :l'utilisateur ne peut pas redimensionner l'image.L'Image sera affichée dans sa dimension d'origine.

CVSobel [src, dst, xorder, yorder, apertureSize]

- Calculer une dérivée d'une image a une couche par l'operateur de Sobel.
- src (Integer) :. Image source.
- dst (Integer) : Image de destination.
- xorder (Integer) : ordre de la dérivée selon x.
- yorder (Integer) : ordre de la dérivée selon y.
- apertureSize (Integer) : taille du noyau de convolution (1,3,5 ou 7).

Le code source pour filtre sobel

```
#include "opencv2/imgproc/imgproc.hpp"
#include "iostream"

using namespace cv;
using namespace std;

int main( )
{
    Mat src1;
    src1 = imread("lena.jpg", CV_LOAD_IMAGE_COLOR);
    namedWindow( "Original image", CV_WINDOW_AUTOSIZE );
    imshow( "Original image", src1 );

    Mat grey;
    cvtColor(src1, grey, CV_BGR2GRAY);

    Mat sobelx;
    Sobel(grey, sobelx, CV_32F, 1, 0); // On cherche les contours verticaux  x = 1

    double minVal, maxVal;
    minMaxLoc(sobelx, &minVal, &maxVal); //trouver le  minimum et maximum
intensité
    cout << "minVal : " << minVal << endl << "maxVal : " << maxVal <<
endl;

    Mat draw;
    sobelx.convertTo(draw, CV_8U, 255.0/(maxVal - minVal), -minVal *
255.0/(maxVal - minVal));

    namedWindow("image", CV_WINDOW_AUTOSIZE);
    imshow("image", draw);

    waitKey(0);   /* Attendre qu'une touche soit pressée */
    return 0;

        }
```

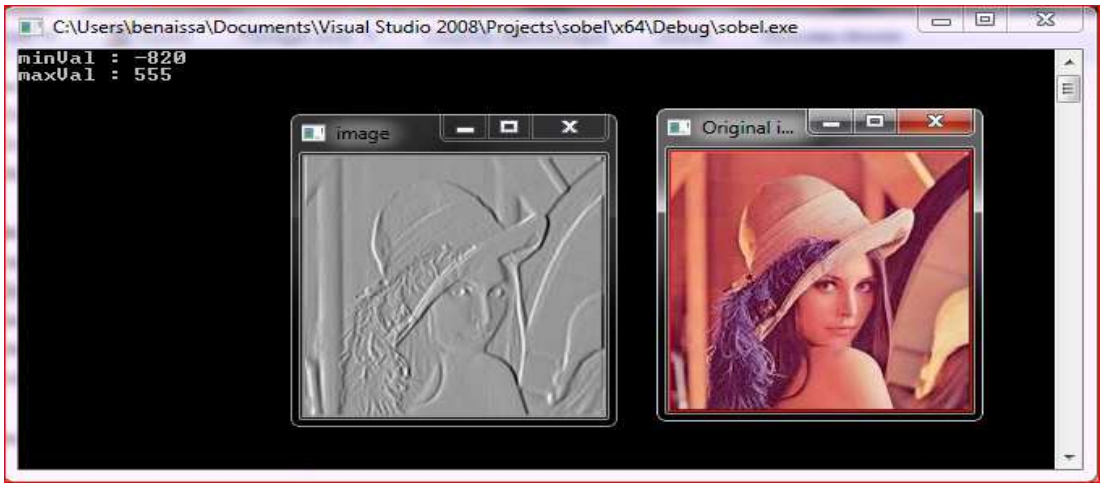


Figure IV. 4 : filtrage d'une image (filtre de sobel).

3.5. La Segmentation par seuillage:

❖ **Pour les vidéos**

La méthode inrange

Cette fonction est composée de 3 arguments elle va permettre de détecter la couleur entre une Limite basse et une limite haute.

Pour choisir un intervalle plutôt qu'un seuil, la fonction inRange peut être préférée.

- **void inRange**(InputArray src, InputArray lowerb, InputArray upperb,OutputArray dst);

  - ✓ src : image à traiter
  - ✓ lowerb : scalaire qui définit le plus petit seuil.
  - ✓ upperb : scalaire qui définit le plus grand seuil.
  - ✓ dst : image de destination au format CV_8U.

- La fonction **cvCreateTrackbar** : La fonction créé le curseur et l'attache à une fenêtre spécifié.

- **int cvCreateTrackbar**(const char* trackbar_name, const char* window_name, int* value, int count, CvTrackbarCallback on_change);

Paramètres

trackbar_name : nom du curseur créé

window_name : nom de la fenêtre qui sera utilisé comme parent pour créer le curseur.

Value : pointeur sur une variable entière d'ont la valeur est l'image de la position du curseur

count : Position maximal du curseur

on_change : pointeur sur la fonction qui sera appelée toutes les fois que le curseur changera de position

Cette fonction aura comme prototype

void Foo(int); peut être NULL si l'appel n'est pas nécessaire.

- La fonction threshold est très utile pour discriminer en une dimension.

Le code source de segmentation par seuillage

```
#include <iostream>
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"

using namespace cv;
using namespace std;

 int main( int argc, char** argv )
 {
    VideoCapture cap(0); //capture de video a partir de webcam

    if ( !cap.isOpened() )
    {
         cout << "Cannot open the web cam" << endl;
         return -1;
    }

    namedWindow("Control", CV_WINDOW_AUTOSIZE);

 int iLowH = 30;
 int iHighH = 100;

 int iLowS = 30;
 int iHighS = 100;

 int iLowV = 30;
 int iHighV = 100;


 cvCreateTrackbar("LowH", "Control", &iLowH, 30); //Hue (0 - 179)
 cvCreateTrackbar("HighH", "Control", &iHighH, 100);

 cvCreateTrackbar("LowS", "Control", &iLowS, 30); //Saturation (0 - 255)
 cvCreateTrackbar("HighS", "Control", &iHighS, 100);

 cvCreateTrackbar("LowV", "Control", &iLowV, 30); //Valeur (0 - 255)
```

```
 cvCreateTrackbar("HighV", "Control", &iHighV, 100);

    while (true)
    {
        Mat imgOriginal;

        bool bSuccess = cap.read(imgOriginal);

         if (!bSuccess)
        {
             cout << "impossible d'ouvrir le webcam" << endl;
             break;
        }

   Mat imgHSV;

  cvtColor(imgOriginal, imgHSV, COLOR_BGR2HSV);

  Mat imgThresholded;


  inRange(imgHSV, Scalar(iLowH, iLowS, iLowV), Scalar(iHighH, iHighS,
iHighV), imgThresholded);


  erode(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(2, 2)) );
  dilate( imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(2, 2)) );


  dilate( imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(2, 2)) );
  erode(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(2, 2)) );

  imshow("SEGMENTATION Image", imgThresholded);
  imshow("Original", imgOriginal);


            if (waitKey(30) == 27)
       {
            cout << "esc pour quitter" << endl;
            break;
       }
    }

   return 0;
}
```

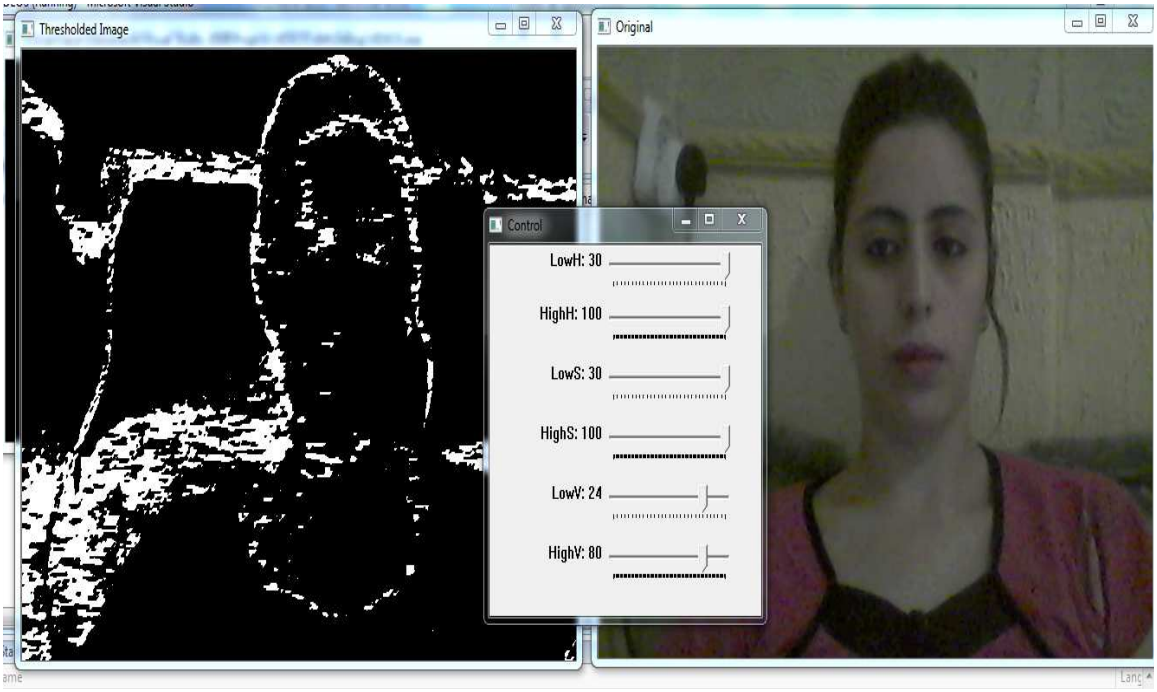


Figure IV. 5 : segmentation des vidéos

4. Conclusion

Dans ce chapitre, nous avons présentées les différents résultats de notre application qui concerne les différents traitements des images et des vidéos.

Ainsi nous avons expliqué les fonctions et les opérations les plus utilisées dans la notre application.

Conclusion général

Conclusion générale

Tout au long de la préparation de notre projet de fin d'étude, nous avons essayé de mettre en pratique les connaissances acquises durant nos études universitaires et cela dans le but de réaliser une application capable de traitement d'image et vidéo.

Au cours de cette mémoire, nous nous étudiés et implémentés les différent algorithmes de segmentation, filtrage et détection de contour appliqué dans le traitement des images et vidéos.

Le traitement image se place après les étapes d'acquisition et de numérisation assurant les transformations d'images et la partie de calcul permettant d'aller vers une interprétation des images traité. Cette phase d'interprétation est d'ailleurs de plus en plus intégrée dans le traitement d'images.

La bibliothèque open cv est un élément important dans la réalisation et de développement de notre projet et cela revient à la richesse de différentes fonctions qui le compose surtout dans le traitement des images et des vidéos.

Toutes fois, un vaste champ de recherche reste ouvert aux futurs chercheurs, qui voudraient faire le suivi de traitement d'image te vidéo aussi l'amélioration les conditions de capture de vidéo en utilisant les outils que nous offre la bibliothèque d'OpenCV.

Tout travail humain étant sujet à des imperfections, nous disons bienvenues aux suggestions et remarques d'autres chercheurs qui pourront élargir de plus les frontières de la science.

Bibliographie

[1] : Alexandre Topol, OpenCV & FFmpeg, Spécialité Informatique Conservatoire National des Arts & Métiers 2009-2010

[2] : Frédéric Dufaux , Traitement du Signal et des Images TELECOM ParisTech, 26 janvier 2011.

[3] : Patrick Gros, traitement de l'image et de la vidéo, 2014

[4] : Stéphane Bazeille, Traitement d'images et vision par ordinateur avec la librairie OPENCV.2012

[5] : Frederic Legrand, Interface Mathematica pour OpenCV, 2010.

[6] : Gilles Boulet, introduction à la vidéo numérique, 2010

[7] : OPENCV , le traitement d'image,2012

[8] : Lingrand Diane , cours de traitement d'images, Projet RAINBOW, Rapport de recherche ,ISRN I3S/RR-2004-05-FR,22 Janvier 2004

[9] : Vision et traitement d'images embarqué,TP 3&4 Lire et traiter le flux d'une Webcam avec OpenCV 2012-2013

[10] : François Meunier DMI, OpenCV: Introduction-2 au Traitement d'Images et Vidéo

[11] : Marc-André Gardner, Introduction a OpenCV, Design III : Intégration Département de génie électrique, génie informatique Faculté des sciences et de génie Université Laval, Hiver 2014

[12] : Mahdi Rezaei, Department of Computer Science, the University of Auckland, Face and Eye Detection Using OpenCV, 2013

[13] : Robert Laganière, OpenCV 2 Computer Vision Application Programming Cookbook, 2012.

[14] : OpenCV , ESIEE Paris, INF-4101/ R. Kachouri, INF-2013-2014

[15] : Samuel ROUXEL , Outils et solutions libres pour la R&D,OPENCV,1er juillet 2010

[16] : Kenneth Dawson-Howe , A PRACTICAL INTRODUCTION TO COMPUTER VISION WITH OPENCV

webgraphie

[17]:http://www.memoireonline.com/01/14/8585/m_Identification-des-personnes-par-reconnaissance-de-visage-pour-la-securite-d-une-institution-banca19.html

[18]: http://fr.openclassrooms.com/informatique/cours

[19]:http://fr.wikipedia.org/wiki/

[20]: www.kaddour.com/chap1/chap1.html

[21] :www.mysic-assoc.net/spip/IMG/pdf/te-traitement-d-image-2.pdf

[22] : http://images.math.cnrs.fr/Le-traitement-numerique-des-images.html

[23]: http://www.memoireonline.com/

[24]: http://www.mycube.fr/quest-ce-quune-image-numerique